

分类号:

学校代号: 11845

UDC:

密级:

学 号: 2110704250

广东工业大学硕士学位论文

(工学硕士)

基于 LabVIEW 的多轴运动控制器应用研究

梁永忠

指导教师姓名、职称: 邓则名教授 高军礼副教授

企业导师姓名、职称: 无

专业或领域名称: 控制理论与控制工程

学生所属学院: 自动化学院

论文答辩日期: 2010年6月6日

A Dissertation Submitted to Guangdong University of
Technology for the Degree of Master of Engineering Science

The application research on multi-axis motion
controller based on LabVIEW

Master Degree Candidate: Liang Yongzhong

Supervisor: Prof.Deng Zeming

Associate Prof.Gao Junli

June 2010
Faculty of Automation
Guangdong University of Technology
Guangzhou, Guangdong, P. R. China, 51000

摘 要

运动控制是 20 世纪 90 年代在国际上兴起的结合现代电力电子技术、计算机技术、传感器技术等进行控制系统设计的一门多学科交叉的技术,在数控机床、汽车、轻工、纺织和军事等领域应用广泛,其中的数控技术、机器人技术更是一个国家运动控制技术发展水平的重要标志。

虚拟仪器技术就是利用高性能的模块化硬件,结合高效灵活的软件来完成各种测试、测量和自动化的应用,目前在测控领域应用广泛,在控制领域却研究得不多。课题旨在研究在 LabVIEW 开发环境下,结合实验室现有的硬件设备开发一套三维运动控制系统。主要研究内容包括:系统总体方案设计、运动控制理论研究、硬件电路研究、软件设计和系统验证等。

系统总体方案设计:采用“PC+运动控制卡”构成多轴运动控制器的控制策略,实现将上位机和下位机分层控制,充分发挥两者的优势——PC 强大的数据处理能力,运动控制卡对电机的精确控制,大大地提高了系统的可靠性和精确性。

运动控制理论研究:比较分析数值插补算法、加/减速控制算法及 NURBS 曲线插补等。数值插补算法包括逐点比较插补法、数字积分插补法和时间分割插补法;加减速控制算法包括梯形和 S 形加/减速控制算法。

硬件电路研究包括运动控制卡的硬件结构分析、运动轴的限位及回零方式设计和控制卡与驱动器之间的连接等。

软件设计:基于虚拟仪器开发环境 LabVIEW 8.6,设计人机控制界面和程序,实现运动轨迹的仿真和对电机的精确控制。另外增加了系统的扩展功能,如 PWM 输出, I/O 控制等。

系统验证:通过在有机玻璃加工仿真的轮廓图案,分析其存在的误差并找出误差来源及改进措施。

关键词: 运动控制; 三维运动控制系统; 运动控制卡; LabVIEW

Abstract

The motion control is a multi-disciplinary overlapping technology internationally emerged in the 1990s, which combines the modern electric power and electronic technology, the computer technology, the sensor technology and so on to carry on the control system's design; it's widely used in the NC machine tool, automobile, light industry, textiles and military fields, in which the numerical control technology and the robot technology are an important sign of the country's level of development of motion control technology.

Virtual instrument technology is the use of high-performance modular hardware, combined with efficient and flexible software to accomplish a variety of test, measurement and automation applications, and now is widely used in the measurement and control areas, but little in the control area. The paper aims to develop a three-dimensional motion control system based on LabVIEW, combine with the existing laboratory hardware. The main research contents include system overall concept design, motion control theory, hardware circuit, software design and system verification and so on.

System overall concept design: use of "PC + motion control card" form multi-axis motion controller's control strategy to realize the host computer and slave hierarchical control, display their superiority — PC powerful data processing capability, motion control card on the precise motor control, greatly improves the system's reliability and accuracy.

Research on motion control theories: comparison and analysis the numerical interpolation algorithms, acceleration and deceleration control algorithms and NURBS curve interpolation. Numerical interpolation algorithms include point by point comparison interpolation, digital integral interpolation and time-sharing interpolation; acceleration and deceleration control algorithms include trapezoidal and S-curve control algorithm.

Hardware circuit research includes the analysis of the motion control card's hardware structure, the design of motion axes' limitation and back to zero, the

connection between the control card and the drivers and so on.

Software design: design the man-machine interactive interface and control procedures based on virtual instrument development environment LabVIEW 8.6 to achieve trajectory simulation and precise control of the motor. Moreover, it's increased the system's extended functions, such as the PWM output, I/O control and so on.

System verification: processes the outline design of simulation on the plexiglass, analyses and identify the source of error and improvement.

Keywords: motion control; three-dimensional motion control system; motion control card; LabVIEW

摘 要	I
Abstract	II
目 录	V
Contents	VIII
第一章 绪论	1
1.1 课题来源	1
1.2 运动控制技术的发展现状	1
1.2.1 国外运动控制技术的发展现状	1
1.2.2 国内运动控制技术的发展现状	2
1.3 虚拟仪器简介及发展现状	2
1.4 课题研究意义	3
1.5 主要研究内容	4
1.6 本章小结	4
第二章 系统总体方案设计	5
2.1 运动控制系统方案选择	5
2.2 系统方案设计	6
2.3 运动控制卡选择	6
2.4 电机及其驱动器选择	7
2.5 上位机控制软件选择	9
2.6 本章小结	10
第三章 运动控制相关理论研究	11
3.1 常用数值插补算法	11
3.1.1 逐点比较插补法	11
3.1.2 数字积分插补法	13
3.1.3 时间分割插补法	15
3.1.4 数值插补算法比较及选择	17
3.2 NURBS 曲线插补研究	17
3.2.1 NURBS 曲线数学模型	17

3.2.2 NURBS 曲线计算.....	18
3.2.3 轮廓误差分析与控制.....	19
3.2.4 法向加速度分析与控制.....	20
3.3 速度控制.....	20
3.3.1 梯形曲线加减速控制算法.....	21
3.3.2 指数加减速控制算法.....	22
3.3.3 S 曲线加减速控制算法.....	23
3.3.4 速度控制算法比较及选择.....	25
3.4 本章小结.....	26
第四章 系统硬件设计.....	27
4.1 控制卡工作原理及结构分析.....	27
4.1.1 电源模块.....	28
4.1.2 主控单元.....	28
4.1.3 主机通信接口.....	29
4.1.4 控制卡与 PC 机的通信模块.....	30
4.1.5 通用数字 I/O 信号.....	31
4.1.6 PWM 产生及模拟量输出.....	32
4.1.7 电机控制模块.....	33
4.1.8 编码器模块.....	34
4.2 行程限位保护.....	35
4.3 运动轴回零.....	36
4.4 驱动器与电机连接.....	37
4.5 整体连接.....	38
4.6 抗干扰措施.....	40
4.7 本章小结.....	40
第五章 系统软件设计.....	41
5.1 LabVIEW 简介.....	41
5.2 LabVIEW 应用程序开发.....	42
5.2.1 LabVIEW 应用程序构成.....	42
5.2.2 建立子 VI.....	42

5.2.3 LabVIEW 应用程序开发步骤	43
5.3 控制卡基于 LabVIEW 驱动研究	44
5.3.1 动态链接库简介	44
5.3.2 调用库函数节点技术	45
5.4 人机控制界面	48
5.5 程序实现	50
5.5.1 仿真程序实现	50
5.5.2 控制程序实现	55
5.6 LabVIEW 程序打包	58
5.7 本章小结	59
第六章 系统验证	60
6.1 仿真待加工的图案	60
6.2 加工已仿真的图案	61
6.3 实验分析总结	61
6.4 本章小结	62
结论与展望	63
参考文献	64
攻读硕士学位期间发表的论文	67
独创性声明	68
致 谢	69

Contents

Abstract(Chinese)	I
Abstract(English)	II
Contents(Chinese)	V
Contents(English)	VIII
Chapter 1 Preface	1
1.1 Subject origin	1
1.2 Development of motion control technology.....	1
1.2.1 Overseas development of motion control technology	1
1.2.2 Domestic status of motion control technology.....	2
1.3 Brief introduction and development of virtual instruments	2
1.4 Significance of the research	3
1.5 Main research contents	4
1.6 Summary	4
Chapter 2 System overall concept design	5
2.1 Selection of motion control scheme	5
2.2 System scheme design.....	6
2.3 Selection of motion control card	6
2.4 Selection of motor and its driver	7
2.5 Selection of PC software	9
2.6 Summary	10
Chapter 3 Analysis of motion control theories	11
3.1 Common interpolation algorithm	11
3.1.1 Point by Point Comparison Interpolation.....	11
3.1.2 Digital Integral Interpolation.....	13
3.1.3 Time-Sharing Interpolation	15
3.1.4 Comparison and selection of interpolation algorithms.....	17
3.2 NURBS curve interpolation research	17
3.2.1 Mathematical model of NURBS curve	17
3.2.2 Calculation of NURBS curve.....	18

3.2.3 Analysis and control of contour error	19
3.2.4 Analysis and control of normal acceleration	20
3.3 Speed control	20
3.3.1 Trapezoidal curve Acc and Dec control Algorithm	21
3.3.2 Exponent Acc and Dec control algorithm	22
3.3.3 S-curve Acc and Dec control algorithm	23
3.3.4 Comparison and selection of speed control algorithms.....	25
3.4 Summary	26
Chapter 4 System hardware design	27
4.1 Controller working principle and structure analysis	27
4.1.1 Power module.....	28
4.1.2 Controller unit	28
4.1.3 Host communication interface.....	29
4.1.4 Communication module between controller and PC.....	30
4.1.5 General digital IO signals.....	31
4.1.6 PWM generation and analog output.....	32
4.1.7 Motor control module.....	33
4.1.8 Encoder module.....	34
4.2 Travel limit protection	35
4.3 Movement axis back to zero.....	36
4.4 Driver and motor connection.....	37
4.5 Whole connection.....	38
3.6 Anti-interference measures.....	40
3.7 Summary	40
Chapter 5 System software design	41
5.1 LabVIEW introduction.....	41
5.2 LabVIEW application program development	42
5.2.1 Constitution of LabVIEW application program.....	42
5.2.2 Establishment of sub-VI.....	42
5.2.3 LabVIEW application program development steps	43

5.3 Driven research of controller based on LabVIEW	44
5.3.1 Dynamic Link Library Introduction	44
5.3.2 Call Library Function Node technology	45
5.4 Man-machine control interface	48
5.5 Program implementation	50
5.5.1 Simulation program implementation	50
5.5.2 Control program implementation	55
5.6 LabVIEW program release	58
5.7 Summary	59
Chapter 6 System verification	60
6.1 Simulating the pattern to be processed	60
6.2 Processing the pattern has been simulated	61
6.3 Analysis and summarize of the experiment	61
6.4 Summary	62
Conclusions and Outlook	63
References	64
Publication during studing for master's studies	67
Originality statement of academic dissertation	68
Acknowledgement	69

第一章 绪论

运动控制是对机械运动部件的位置、速度、加速度和轨迹控制等进行实时的控制管理,使其按照预期的运动轨迹和规定的运动参数进行运动^[1]。早期的运动控制技术主要是伴随着数控技术、机器人技术和工厂自动化技术的发展而发展的,经过多年的发展,现代运动控制已成为一个以控制理论为基础,涵盖电机技术、电力电子技术、传感器检测技术、信息处理技术、自动控制技术、计算机仿真和辅助制造技术等许多学科交叉应用的控制技术^[2]。运动控制技术不仅应用于数控机床、工业机器人、轻工、纺织、化工等传统行业,而且还在火炮操纵、雷达驱动、导弹发射、导航控制等多个领域得到了广泛的应用,其中的数控技术、机器人技术更是一个国家运动控制技术发展水平的重要标志。

1.1 课题来源

来源于粤港关键领域重点突破基金项目:多轴数控超声加工成套设备,编号:2006A15101004。本课题属于该项目的前期研究,主要在系统方案及软硬件设计方面作初步探讨。

1.2 运动控制技术的发展现状

1.2.1 国外运动控制技术的发展现状

早期的运动控制器无需另外的处理器和操作系统的支持,就能独立完成运动控制功能,是可以独立运行的专用的控制器。这类运动控制器一般已根据应用行业的工艺要求设计了相关的功能,所以不能离开其特定的工艺要求而跨行业应用,用户也不能根据应用要求来重组自己的运动控制系统。这种运动控制器结构封闭,功能单一,软件移植性、组网通讯能力差,不能满足现代工业和社会发展的要求。针对这种情况,1987年起,世界各国开始研究一种开放式体系结构的数字运动控制系统,即系统要具有互换性、可伸缩性、可移植性、互操作性和可扩展性。首先是美国空军在美国政府资助下发表了著名的下一代控制器NGC(Next Generation Workstation/Machine Controller)研究计划,该计划重要内容之一便是提出了“开放系统体系结构

标准规格(OSACA)”,目的是要建立一种开放式的控制系统标准,使不同的设计人员可依据规范开发具有互换性和互操作性的控制器部件,打破少数制造商的技术垄断,加快控制系统的发展步伐。其后有许多相关的研究计划在世界各国相继启动,其中影响较大的有美国的开放式模块化结构控制器OMAC (Open Modular Architecture Controller)、欧洲的OSACA(Open System Architecture for Control within Automation)和日本的OSEC(Open System Environment for Controller Architecture)等计划^[9]。

1.2.2 国内运动控制技术的发展现状

“八五”期间,我国广大科研工作者成功地开发了两种数控平台和华中I型、蓝天I型、航天I型、中华I型等4种数控系统,这些系统采用模块化、嵌入式的软、硬件结构。其中以华中I型较具代表性,它采用工业PC机上插接口卡的结构,运行在DOS平台上,具有较好的模块化、层次化特点,具有一定的扩展和伸缩性。但从整体上来说,这些系统还不具备开放性系统的本质特征^[10]。

近年来,随着运动控制技术的不断进步和完善,运动控制器作为一个独立的工业自动化控制类产品,已经被越来越多的产业领域接受,并且已经达到一个引人注目的市场规模。我国在运动控制器及运动控制系统开发方面相对滞后,1999年固高科技有限公司开始开发、生产开放式运动控制器及运动控制系统,随后,国内又有其他几家公司相继进入该领域,如摩信、北京东方嘉宏等。由此可见,我国在运动控制技术的研究和应用方面与美国等发达国家相比还有较大差距。

1.3 虚拟仪器简介及发展现状

虚拟仪器技术就是利用高性能的模块化硬件,结合高效灵活的软件来完成各种测试、测量和自动化的应用。其基本思想是利用计算机来管理仪器、组织仪器系统,进而逐步代替仪器完成某些功能,最终达到取代传统电子仪器的目的。

1986年,美国国家仪器公司(National Instrument)首先提出基于计算机技术的虚拟仪器概念,研制和推出了基于多种总线系统的虚拟仪器,其中的LabVIEW应用最为广泛,随后HP公司推出了HPVVEE、HPTIG图形化开发工具,提供数十至数百种虚拟仪器的组建单元和整机,用户可以用它组建或挑选自己所需的仪器,给虚拟仪器的开发提供了极为方便的开发环境。除此之外,世界上陆续有数百家公司,如Tektronix公司, Racal公司、HEM Data公司等也相继推出了总线系统多达数百个品种的虚

拟式仪器^[5,9]。从那以后,“软件即是仪器”的概念逐步形成,虚拟仪器的设想变为现实。与传统仪器相比,虚拟仪器基于计算机的开放系统,可方便地同外设、网络及其它应用连接,价格低廉,可重用使用,技术更新快,基于软件体系的结构,大大节省开发维护费用。

目前虚拟仪器在测试和测量领域应用广泛,如美国斯坦福大学的虚拟仪器教学、实验、仿真系统;挪威CARDIAC公司的基于LabVIEW平台的测试北海油田石油、大气、水流的MPFM等系统;清华大学应用虚拟仪器技术构建的用于检测汽车发动机性能的出厂检测系统等。发展到今天,虚拟仪器技术已不再局限于测控领域了,NI率先将这一技术扩展到控制部分,使得虚拟仪器在控制领域的应用也越来越多。如Texas大学使用Labview,Lab-PC-1200及GPIB卡开发了一个自动控制系统实验室及一个风洞电压控制系统,使学生可以在实验室得到更深入、更实际的自动控制系统开发经验;Sorin Grama, Eric T. Lee等人使用LabVIEW、PCI-E系列数据采集卡完成了一个检测心脏助搏器中血液泵可靠性的闭环控制系统。采用Labview中的多线程技术通过数据采集卡同时控制4个血液泵,该系统可连续工作45天以上,具有很高的稳定性和可靠性。

但是,目前虚拟仪器在控制领域的应用主要集中在国外,在国内的应用还不多,很少研究人员在此领域进行相关的研究应用。

1.4 课题研究意义

主要有以下两点:1、运动控制技术在数控加工、工业生产、军事、航空航天等众多领域应用广泛,其中的数控技术是现代制造业的核心技术,是衡量一个国家制造水平的重要标志之一。我国是一个制造业大国,但数控技术水平还不高,与欧美、日本等制造业强国还有相当大的差距,为此,对运动控制技术的研究是相当有必要的,本文针对这种情况,结合虚拟仪器软件LabVIEW,在运动控制方面作初步探讨,开发一套开放式的三维运动控制系统,该系统不仅具有空间直线、圆弧、螺旋线插补功能,还具有NURBS曲线插补功能。

2、LabVIEW是美国NI公司开发的图形化编程环境,由于其具有强大的界面编辑、数据采集和信号处理等功能,在测控领域应用非常广泛,国内外也有不少专家学者在这方面作过研究。然而,通过查阅相关资料,国内只有极少数研究人员采用LabVIEW在运动轨迹仿真及控制方面作过研究。因此,本课题采用LabVIEW作为三

维运动控制系统的开发软件具有创新价值，可为以后进一步的开发研究奠定基础。

1.5 主要研究内容

课题所研究的三维运动控制系统能在LabVIEW下仿真待加工物体的轮廓特征，并结合机械平台加工。此外，通过运动控制卡的I/O口可以控制外部设备。论文完成的主要内容如下：

- 1、研究运动控制技术及虚拟仪器技术在国内外的发展现状，分析研究常用的几种运动控制方案，选择适合于本系统的设计方案。
- 2、分析设计系统各部分硬件电路，搭建控制平台等。
- 3、研究运动控制理论，包括常用的数值插补算法、加减速控制算法及NURBS曲线插补等。
- 4、基于LabVIEW设计人机控制界面，开发运动仿真程序及控制程序。
- 5、通过在有机玻璃上加工仿真图案来验证系统，并将其与仿真图案对比，做误差分析，找出系统存在的不足及改善措施。

1.6 本章小结

概述了运动控制技术研究的重要性，阐述了国内外运动控制技术及虚拟仪器技术的发展现状，阐明了课题的研究来源及意义，并说明了课题的主要研究内容。

第二章 系统总体方案设计

高性能、高可靠性、低成本通常是每一个开发人员开发一套设备遵从的原则之一。目前, 控制技术在数控加工、纺织等诸多领域应用广泛, 其竞争也非常激烈。因此, 在作进一步研究之前需先确定系统的总体方案, 课题的目的是开发一套三维运动控制系统, 即 X、Y 和 Z 三维, 此外, 通过在有机玻璃上雕刻仿真的图案来验证系统, 还需加一个电主轴。

2.1 运动控制系统方案选择

运动控制技术应用广泛, 不同的领域、场合使用不同的控制方案, 常用的运动控制系统方案有如下几种^[1]:

(1) 单片机系统

采用单片机作为控制器的控制系统, 优点是成本较低, 缺点是开发难度较大, 需要开发人员对运动控制领域比较熟悉, 开发周期长, 而且由于一般的单片机受运算速度的限制, 无法实现实时性要求较高的、复杂的控制算法。所以此方案只适用于产品批量较大、控制功能简单、实时性要求不高的系统。

(2) 专业运动控制 PLC

PLC 是计算机家族中的一员, 是为工业控制应用而设计制造的。此方案一般适用于运动过程比较简单、运动轨迹固定的设备, 如送料系统、自动焊机等。

(3) 软 CNC

在 PC 上编写系统程序, 配上 I/O 接口卡输出脉冲和方向信号来控制步进或伺服电机。受微机定时器的限制, 最大脉冲频率只有 100K 左右, 在控制伺服电机时会有速度和精度的问题。而且 Windows 环境下由于其多任务的机制, 若没有深入 Windows 内核进行底层编程来发脉冲, 难以保证控制的精确性。

(4) PC 和运动控制卡

PC 和运动控制卡构成多轴运动控制器, 运动控制卡只需要从 PC 接收控制命令, 然后自己完成运动有关的控制, 几乎不占用微机 CPU 时间。随着 PC 的普及, 此方案将是运动控制系统的一个主要的发展方向。而且大多数运动控制卡提供了 Windows 环境下的动态链接库, 使用非常方便, 不仅大大缩短产品开发周期, 而且

能够实现更完善的运动控制系统。

从以上可以看出，“PC+运动控制卡” 构成多轴运动控制器的控制方案将会成为主流，所以本系统采用“PC+运动控制卡”的主从控制模式设计多轴运动控制系统。

2.2 系统方案设计

本课题要开发一套三维运动控制系统，其方案设计图如下图2-1所示。PC与运动控制卡之间采用PCI总线的方式通讯，运动控制卡与电机驱动器及外部信号使用光耦隔离接口板连接，本系统可以使用步进电机或伺服电机，步进电机一般用于开环控制的经济型系统中，伺服电机主要应用在需要精确位置控制的闭环控制系统中，受到目前条件限制，使用了步进电机。

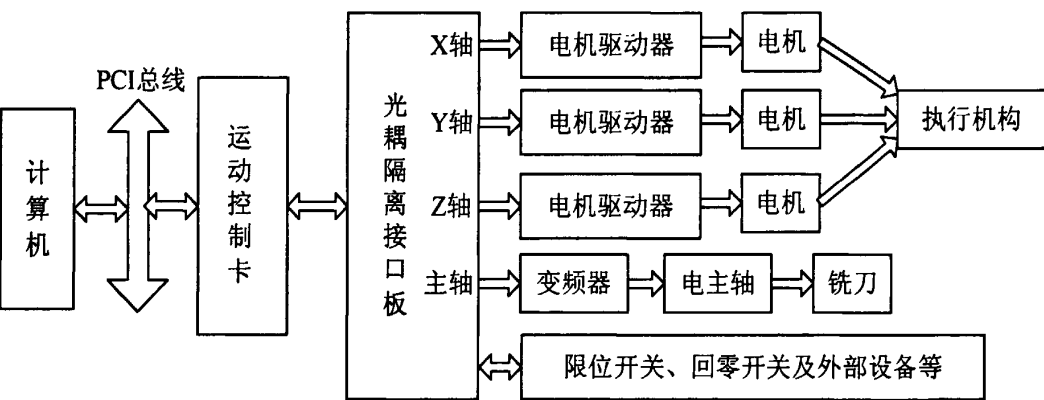


图 2-1 系统方案设计图

Figure2-1 System Design Diagram

PC 负责人机控制界面的管理和控制系统的实时监控等方面的工作^[4]；控制卡完成运动控制的所有细节(包括脉冲和方向信号的输入输出、自动加减速的处理、原点和限位等信号的检测等等)；电机驱动器接收运动控制卡的指令，驱动电机，由电机带动机械平台；回零开关和限位开关用于确定准确的物理坐标及保护轴运动行程。

2.3 运动控制卡选择

本系统需要控制三个运动轴及一个主轴，运动轴的控制一般采用数字脉冲控制的方式，主轴的控制一般采用模拟量控制的方式，所以选择的运动控制卡至少能控制3轴及输出一路模拟量。

目前生产运动控制卡的公司比较多，国外比较著名的有Delta Tau、Trio、Adlink、

NI等，国内的有研华、摩信、固高、北京东方嘉宏等，由于北京东方嘉宏公司生产的6050四轴运动控制卡与同类产品相比具有较高的性价比，尤其在控制精度、速度和易操作性方面表现优异，所以本系统采用6050四轴运动控制卡，如下图2-2所示。

6050四轴运动控制卡是基于PC机PCI总线的步进电机或数字式伺服电机的上位控制单元，具有4轴直线插补/4轴圆弧插补（任意2轴圆弧+2轴直线）功能，能够实现最多4卡共用板间插补，支持梯形、S形加减速，提供20路光隔输入，8路光隔/达林顿放大输出，1路PWM脉冲/12位DA模拟量输出，PWM输出频率范围为18-150000Hz，DA输出分辨率为12位，输出电压范围为-10VDC至+10VDC，提供的运动控制函数库支持Windows98/2000/XP环境下VB、VC和Delphi等编程。

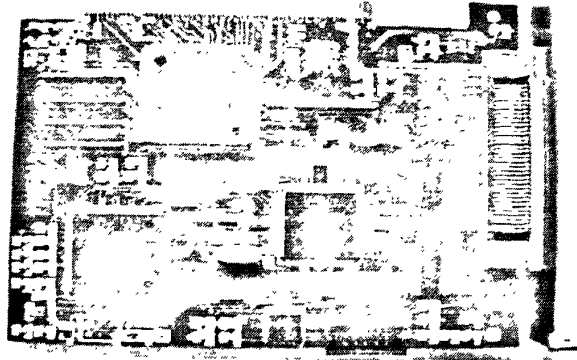


图 2-2 运动控制卡

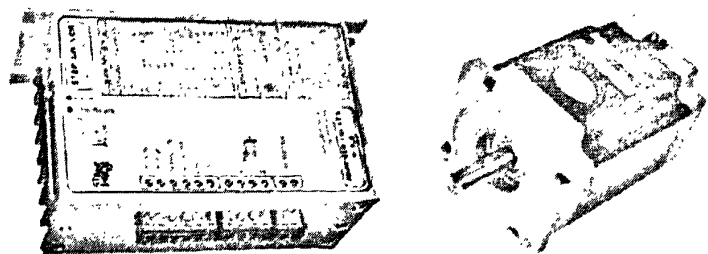
Figure2-2 Motion Control Card Diagram

2.4 电机及其驱动器选择

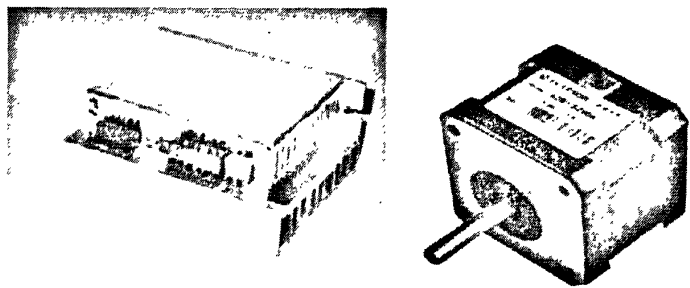
1、运动轴

本系统有三个运动轴，即X、Y和Z三轴。其中X轴和Z轴的步进电机主要带动电主轴运动，所需转矩大约0.2Nm，而Y轴的步进电机要带动加工平台上整个支架运动，所需转矩大约2.5Nm。从系统性能、驱动器及电机的性价比和本实验室情况考虑，Y轴采用了北京斯达微步控制技术公司的34HS300C步进电机及其驱动器，X轴和Z轴采用了北京捷科利达机电技术有限公司的JK402M驱动器和北京和利时电机技术有限公司的42BYG两相混合式步进电机。

X、Y和Z三轴步进电机及其驱动器如下图2-3所示，



a)Y 轴步进电机及其驱动器



b) X 轴和 Z 轴步进电机及其驱动器

图 2-3 步进电机及其驱动器

Figure2-3 Diagram of Step Motor and its Driver

Y 轴驱动器及电机的主要性能指标分别如下表 2-1 及表 2-2 所示：

表 2-1 Y 轴驱动器主要性能指标

Table 2-1 Driver main performance index of Y-axis

供电电源	AC40V，-15%—10%，功率不小于 140W
输出电流范围	0.4A-5.0A，16 档可调
步进角	16 档可调
相数	2 相混合式

表 2-2 Y 轴步进电机主要性能指标

Table 2-1 Step motor main performance index of Y-axis

相电流	5.0A
最大静转矩	4.0Nm
相电阻	0.75 Ω
相电感	7.0mH
空载启动转速	252 转/分
步距角	0.9° /1.8°

X轴和Z轴驱动器及电机的主要性能指标分别如下表2-3及表2-4所示：

表 2-3 X、Z 轴驱动器主要性能指标

Table 2-3 Driver main performance index of XZ-axis

供电电源	DC24V
输出电流范围	0.7A-3.0A，8 档可调
步进角	8 档可调
相数	2 相混合式

表 2-4 X、Z 轴步进驱动器主要性能指标

Table 2-4 Step motor main performance index of XZ-axis

静态相电流	1.5A
保持转矩	0.43Nm
相电阻	1.1 Ω
相电感	5.0mH
空载启动频率	1.5KHz
步距角	0.9° /1.8°

2、主轴

主轴控制使用了欧瑞传动电气有限公司的F2000-M0015S2B变频器及江苏星晨公司的高速电主轴，如下图2-4所示，变频器具有端子控制、操作面板控制和串行通讯口控制三种方式，高速电主轴的额定频率是133-400Hz，额定转速是8000-24000rpm。

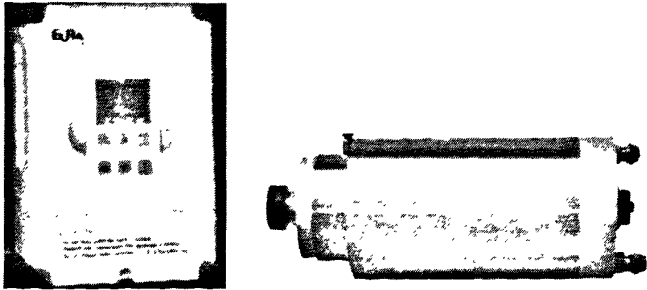


图 2-4 变频器及电主轴

Figure2-4 Diagram of Converter and Electric Spindle

2.5 上位机控制软件选择

上位机控制软件可以利用VB、VC、Delphi等多种编程语言编写，而且6050运动

控制卡也提供了其相应的驱动程序,在这些编程环境中很容易实现所需的运动控制,但是,如果要开发出一个美观的人机控制界面,将会有很大一部分时间花在程序主界面的编写上,LabVIEW目前在计算机测控领域应用广泛,从简单的仪器控制、数据采集到尖端的测控和工业自动化,都可以发现其应用的很多实例,国内外相关学者对此也作了很多的研究工作,发表了不少相关论文,但在运动控制方面,却探讨的不多^[9]。为此,本课题采用LabVIEW作为上位机控制软件,利用其强大的界面编辑功能以及能控制第三方运动控制卡的特点,在运动控制方面作个研究。

2.6 本章小结

分析对比常用的几种运动控制方案,选择了“PC+运动控制卡”构成多轴运动控制器的方案;从系统性能、硬件的性价比及本实验室的情况等方面考虑,对比选择了运动控制卡、电机及其驱动器、上位机控制软件等。

第三章 运动控制相关理论研究

运动控制卡对电机的控制主要包括控制其位移、速度、加速度等，而实现对这些参数的控制主要用到的算法有数值插补算法、加减速控制算法等，本章将研究这些算法的优缺点。

3.1 常用数值插补算法

在实际应用中，有可能需要几个电机轴按一定轨迹同时运动以完成直线、圆弧或螺旋线运动，这种几个电机轴按一定轨迹同时运动的方式就是所谓的插补运动，这也是数控的核心任务，而数控的其他任务大多是围绕着它进行的。常用的数值插补算法有逐点比较法、数字积分法和时间分割法，下面将对这几种常用的数值插补算法作详细的论述。

3.1.1 逐点比较插补法

逐点比较法是一种边判断边前进，以阶梯折线逼近直线或圆弧等曲线的插补方法。它的基本原理是：数控系统在控制过程中，逐点地计算和判别运动轨迹与给定轨迹的偏差，并根据偏差控制进给轴向给定轮廓靠拢，缩小偏差，使加工轮廓逼近给定轮廓^[10]。

(1) 逐点比较法直线插补

以第一象限为例，设直线的起点坐标为原点，终点坐标为 $A(X_e, Y_e)$ ，插补点坐标为 $P_i(X_i, Y_i)$ ，如下图3-1所示。

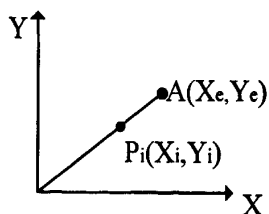


图 3-1 逐点比较法直线插补

Figure3-1 Linear Interpolation of Point by Point Comparison method

在任意时刻，可以通过比较直线斜率 Y_i/X_i 和 Y_e/X_e 的大小来确定插补点的位置与

加工直线的关系, 根据数控插补原理, 可以构造偏差函数

$$F_i(X_i, Y_i) = Y_i X_e - X_i Y_e \quad (3.1)$$

并且从 $F_i(X_i, Y_i)$ 的数值可以确定下一插补点的位置、进给方向和偏差函数。

若 $P_i(X_i, Y_i)$ 在直线 OA 上方, 则 $F_i(X_i, Y_i) > 0$, 下一步应向 +X 方向进给一步, 新加工点的偏差为

$$F_{i+1} = Y_{i+1} X_e - X_{i+1} Y_e = Y_i X_e - (X_i + 1) Y_e = F_i - Y_e \quad (3.2)$$

进给脉冲总数 n 减 1; 若 $P_i(X_i, Y_i)$ 在直线 OA 下方, 则 $F_i(X_i, Y_i) < 0$, 下一步应向 +Y 方向进给一步, 新加工点的偏差为

$$F_{i+1} = Y_{i+1} X_e - X_{i+1} Y_e = (Y_i + 1) X_e - X_i Y_e = F_i + X_e \quad (3.3)$$

进给脉冲总数 n 减 1; 若 $P_i(X_i, Y_i)$ 刚好在直线 OA 上, 则 $F_i(X_i, Y_i) = 0$, 将此情况归入 $F_i(X_i, Y_i) > 0$, 向 +X 方向进给一步。

(2) 逐点比较法圆弧插补

以第一象限逆圆为例分析逐点比较法圆弧插补原理, 如下图 3-2 所示,

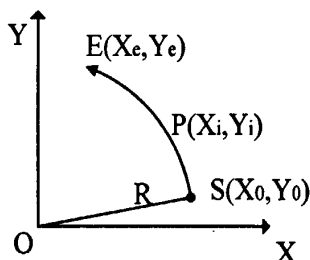


图 3-2 逐点比较法圆弧插补

Figure 3-2 Circular Interpolation of Point by Point Comparison

动点 $P(X_i, Y_i)$ 沿圆心为原点 $O(0,0)$, 半径为 R 的逆圆 SE 运动, 则可设圆弧插补偏差函数表达式为:

$$F_i(X_i, Y_i) = X_i^2 + Y_i^2 - R^2 \quad (3.4)$$

若 $P(X_i, Y_i)$ 在圆外, 则偏差函数 $F_i(X_i, Y_i) > 0$, 为了减小偏差, 应向 -X 轴方向走一步, 新的坐标值为 $X_{i+1} = X_i - 1$, $Y_{i+1} = Y_i$, 新的偏差函数为:

$$F_{i+1}(X_{i+1}, Y_{i+1}) = X_{i+1}^2 + Y_{i+1}^2 - R^2 = (X_i - 1)^2 + Y_i^2 - R^2 = F_i(X_i, Y_i) - 2X_i + 1 \quad (3.5)$$

若 $P(X_i, Y_i)$ 在圆内, 则偏差函数 $F_i(X_i, Y_i) < 0$, 为了减小偏差, 应向 +Y 轴方向走一步, 新的坐标值为 $X_{i+1} = X_i$, $Y_{i+1} = Y_i + 1$, 新的偏差函数为:

$$F_{i+1}(X_{i+1}, Y_{i+1}) = X_{i+1}^2 + Y_{i+1}^2 - R^2 = X_i^2 + (Y_i + 1)^2 - R^2 = F_i(X_i, Y_i) + 2Y_i + 1 \quad (3.6)$$

若 $P(X_i, Y_i)$ 刚好在圆上, 将此情况归入 $F_i(X_i, Y_i) > 0$, 向 -X 轴方向走一步。当

圆弧过象限时,其偏差计算公式和进给方向都会发生改变,这时需作针对性处理,以保证加工继续进行。逐点比较法圆弧插补终点判别的方法不同于直线插补,是利用动点坐标和终点坐标是否重合来判定插补是否结束。

3.1.2 数字积分插补法

(1)数字积分插补法的基本原理

数字积分法又称数字微分分析器(Digital Differential Analyzer,简称DDA)。采用该方法进行插补,具有运算速度快,逻辑功能强,脉冲分配均匀等特点,目前该方法在数控系统中得到广泛的应用^[11, 12]。

其原理是将积分运算转化为求和运算,即平面内任意一条曲线 $X = f(t)$ 在区间 $0 \sim t$ 所包围的面积可用积分求得:

$$S = \int f(t)dt \quad (3.7)$$

当 Δt 取得很小时,如取为最小单位1时,积分运算可用累加求和来近似,则上式(3.7)可变为

$$S = \sum_{i=1}^m x_i \quad (3.8)$$

(2)数字积分法直线插补原理

假设一空间直线OA,起点位于原点,终点为 (X_e, Y_e) ,动点移动的速度为 V ,沿X、Y坐标轴移动的速度分量为 V_x 、 V_y ,如下图3-3所示。

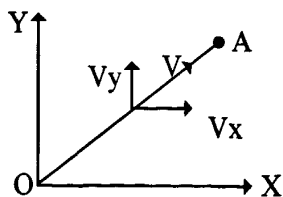


图 3-3 DDA 直线插补

Figure3-3 DDA Linear Interpolation

在 Δt 时间内,动点沿X、Y坐标轴移动的微小增量为

$$\begin{cases} \Delta X = V_x \Delta t \\ \Delta Y = V_y \Delta t \end{cases} \quad (3.9)$$

令 $\frac{V}{OA} = \frac{V_x}{X_e} = \frac{V_y}{Y_e} = k$ ，则在 Δt 时间内，动点沿 X、Y 坐标轴移动的微小增量为

$$\begin{cases} \Delta X = V_x \Delta t = k X_e \Delta t \\ \Delta Y = V_y \Delta t = k Y_e \Delta t \end{cases} \quad (3.10)$$

取 Δt 为最小单位 1，各坐标轴的位移量为下式(3.11)

$$\begin{cases} X = \int k X_e dt = k \sum_{i=1}^m X_e \Delta t = k \sum_{i=1}^m X_e \\ Y = \int k Y_e dt = k \sum_{i=1}^m Y_e \Delta t = k \sum_{i=1}^m Y_e \end{cases} \quad (3.11)$$

所以动点从原点走向终点的过程，可以看作是各坐标轴每经过一个单位时间间隔 Δt ，分别以增量 kX_e 、 kY_e 同时累加的过程。

(3) 数字积分法圆弧插补原理

以第一象限逆圆弧为例，设动点 $P(X_i, Y_i)$ 沿圆弧 SE 运动，轮廓半径为 R ，运动速度为 V ，在 X、Y 两坐标轴上的速度分量分别为 V_x 、 V_y ，如下图 3-4 所示。

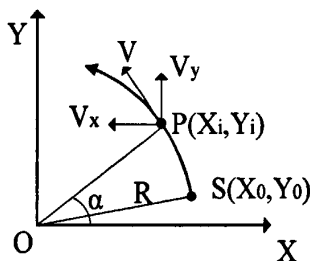


图 3-4 DDA 圆弧插补

Figure 3-4 DDA Circular Interpolation

由圆的参数方程得

$$\begin{cases} X_i = R \cdot \cos \alpha \\ Y_i = R \cdot \sin \alpha \end{cases} \quad (3.12)$$

动点 P 沿 X、Y 轴的速度分量为

$$\begin{cases} V_x = \frac{dX_i}{dt} = -R \cdot \sin \alpha = -Y_i \\ V_y = \frac{dY_i}{dt} = R \cdot \cos \alpha = X_i \end{cases} \quad (3.13)$$

由上式(3.13)可以看出，动点 P 沿 X 轴的速度分量的绝对值等于该点在 Y 方向的瞬时坐标值 Y_i ，沿 Y 轴的速度分量等于该点在 X 方向的瞬时坐标值 X_i ， V_x 、 V_y 是随 P 点

位置的变化而变化的，通过公式变换得

$$\begin{cases} dX_i = -Y_i dt = \Delta X_i = -Y_i \Delta t \\ dY_i = X_i dt = \Delta Y_i = X_i \Delta t \end{cases} \quad (3.14)$$

即在 Δt 时间内，动点沿X、Y轴移动的微小增量分别为 Y_i 、 X_i ，与直线插补刚好相反。

3.1.3 时间分割插补法

时间分割法作为一种数据采样法，是根据用户程序的进给速度，将给定轮廓曲线分割为每一插补周期的进给段，即轮廓步长。每一个插补周期执行一次插补计算，计算出轮廓曲线上的进给段在各坐标轴上的分量大小，计算机定时对坐标的实际位置进行采样，与插补计算的坐标值进行比较，得出位置误差，再根据位置误差对伺服系统进行控制，达到消除误差使实际位置跟随指令位置的目的^[13, 14]。

(1) 时间分割法直线插补

以XY平面第一象限为例分析时间分割法直线插补原理，如下图3-5所示，

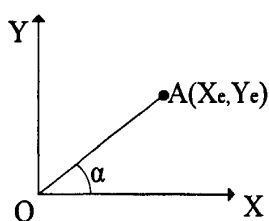


图 3-5 时间分割法直线插补

Figure 3-5 Linear Interpolation of Time-Sharing

设直线段OA起点在原点，终点为 $A(X_e, Y_e)$ ，OA与X轴的夹角为 α 。根据进给速度F得一个插补周期T的进给量

$$\Delta L = FT \quad (3.15)$$

ΔL 在X、Y轴的进给分量分别为

$$\begin{cases} \Delta X = \Delta L \cos \alpha = \Delta L X_e / \sqrt{X_e^2 + Y_e^2} \\ \Delta Y = \Delta L \sin \alpha = \Delta L Y_e / \sqrt{X_e^2 + Y_e^2} \end{cases} \quad (3.16)$$

一般来讲，插补周期T是不变的，因此由进给步长的计算公式(3.15)可以知道进给步长与速度成正比。时间分割法直线插补比较简单，而且是沿着理论直线进给的，

不存在理论上的误差^[16]。

(2)时间分割法圆弧插补

时间分割圆弧插补法也是要根据进给速度F先计算出进给步长 ΔL ，现以XY平面第一象限为例分析其原理，如下图3-6，顺圆弧AB为待加工曲线，弦 \overline{AB} 是圆弧插补每个周期的进给步长 ΔL ，点M是弦 \overline{AB} 的中点。 $\angle COA = \Phi_i$ ， $\angle AOB$ 是弦 \overline{AB} 的圆心角，其值为

$$\angle AOB = \varphi = \frac{\Delta L}{R} = K \quad (3.17)$$

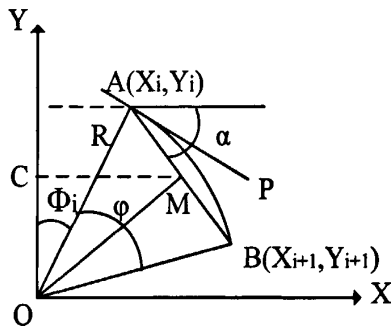


图 3-6 时间分割法圆弧插补

Figure 3-6 Circular Interpolation of Time-Sharing

由图中三角形相似关系可得：

$$\begin{cases} X_i = R \sin \Phi_i \\ Y_i = R \cos \Phi_i \end{cases} \quad (3.18)$$

$$\begin{cases} X_{i+1} = R \sin(\Phi_i + \varphi) = R(\sin \Phi_i \cos \varphi + \cos \Phi_i \sin \varphi) = X_i \cos \varphi + Y_i \sin \varphi \\ Y_{i+1} = R \cos(\Phi_i + \varphi) = R(\cos \Phi_i \cos \varphi - \sin \Phi_i \sin \varphi) = Y_i \cos \varphi - X_i \sin \varphi \end{cases} \quad (3.19)$$

由三角函数的相关知识得

$$\begin{cases} \cos \varphi \approx 1 - \frac{1}{2}\varphi^2 \\ \sin \varphi = \varphi \end{cases} \quad (3.20)$$

则由式(3.17)、(3.19)和(3.20)得

$$\begin{cases} X_{i+1} = X_i(1 - \frac{1}{2}\varphi^2) + Y_i\varphi = X_i \left[1 - \frac{1}{2} \left(\frac{\Delta L}{R} \right)^2 \right] + Y_i \frac{\Delta L}{R} = X_i - \frac{K^2}{2} X_i + K Y_i \\ Y_{i+1} = Y_i(1 - \frac{1}{2}\varphi^2) - X_i\varphi = Y_i \left[1 - \frac{1}{2} \left(\frac{\Delta L}{R} \right)^2 \right] - X_i \frac{\Delta L}{R} = Y_i - \frac{1}{2} K^2 Y_i - K X_i \end{cases} \quad (3.21)$$

因此, 进给量为

$$\begin{cases} \Delta X = X_{i+1} - X_i = KY_i - \frac{K^2}{2} X_i \\ \Delta Y = Y_{i+1} - Y_i = -\frac{1}{2} K^2 Y_i - KX_i \end{cases} \quad (3.22)$$

3.1.4 数值插补算法比较及选择

逐点比较法以一段段微小的折线段来逼近直线或圆弧, 其最大的偏差不超过一个脉冲当量, 其优点是算法简单, 容易用软件实现, 输出脉冲均匀, 而且输出脉冲速度变化小, 进给速度易于控制; 缺点是难以实现两坐标以上的插补, 因此多用于两坐标数控机床。

数字积分法的优点是每个轴的插补对应一个DDA模块, 易于实现三个或以上坐标轴的插补, 而且可以比较容易地实现二次曲线, 甚至高次曲线的插补; 其缺点是对函数的积分运算转化为累加求和运算, 若累加值超过累加寄存器容量, 则输出进给脉冲, 不论加工行程的长短, 都必须完成一定次数的累加运算, 也就是说, 行程长, 进给快; 行程短, 进给慢, 所以它的进给速度难于控制。

时间分割法的优点是用直线段来逼近曲线, 逼近误差比较小, 对直线插补没有逼近误差。插补运算速度与进给速度无严格的关系, 因而采用这类插补算法时, 可达到较高的进给速度; 其缺点是插补的实现算法比较复杂, 插补运算时间比较长^[6]。

从以上分析比较可知, 时间分割法在逼近误差和插补速度方面相对于逐点比较法和数字积分法具有优势。

3.2 NURBS 曲线插补研究

NURBS(Non Uniform Rational B-spline)曲线即非均匀有理B样条曲线, 由于它不仅能够精确地统一表示标准解析曲线和自由曲线, 而且形状控制能力十分强大、灵活, 因此1991年国际标准化组织将其定为工业产品模型数据交换标准。但在数控加工领域, 目前很多CNC机床只支持直线、圆弧插补, 只有少数高档数控系统支持NURBS插补, 所以本系统除了具有空间直线、圆弧、螺旋线插补功能外, 还增加了NURBS曲线插补功能。

3.2.1 NURBS曲线数学模型

一条K次NURBS曲线可以表示为一段有理多项式矢函数,形式如下:

$$P(u) = \frac{\sum_{i=0}^n N_{i,k}(u) \omega_i d_i}{\sum_{i=0}^n N_{i,k}(u) \omega_i} \quad (3.23)$$

式中 $P(u)$ 为曲线上的位置向量; d_i 为控制点,用于确定曲线的位置,通常不在曲线上; w_i 为权因子,它相当于控制点的“引力”,其值越大曲线就越接近控制点; u 为节点矢量, $N_{i,k}(u)$ 为K次样条基函数,按德布尔-考克斯递推公式定义如下:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & , u_i \leq u \leq u_{i+1} \\ 0 & , \text{其它} \end{cases} \\ N_{i,k}(u) = \frac{(u - u_i) N_{i,k-1}(u)}{u_{i+k} - u_i} + \frac{(u_{i+k+1} - u) N_{i+1,k-1}(u)}{u_{i+k+1} - u_{i+1}} & , k \geq 1 \\ \text{规定 } \frac{0}{0} = 0 \end{cases} \quad (3.24)$$

3.2.2 NURBS曲线计算

(1)确定节点矢量

实际应用中,三次NURBS曲线应用较多,所以取 $K=3$,为了使曲线的首末点和首末插值点一致,将首末节点都取重复度 $K+1$ (即为4),即 $u_0 = u_1 = u_2 = u_3$,
 $u_{n+1} = u_{n+2} = u_{n+3} = u_{n+4}$,且在大多数实际应用中,端节点值分别取为0与1。因此,有曲线定义域 $u \in [u_3, u_{n+1}] = [0, 1]$

(2)计算基函数

根据德布尔-考克斯递推公式,将 $k=0,1,2,3$ 依次代入公式,通过递推关系可计算出三次B样条基函数。令 $W_n = u - u_{i+n}$, $T_{mn} = u_{i+m} - u_{i+n}$,得:

0次B样条基函数为

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & , u_i \leq u \leq u_{i+1} \\ 0 & , \text{其它} \end{cases} \\ N_{i+1,0}(u) = \begin{cases} 1 & , u_{i+1} \leq u \leq u_{i+2} \\ 0 & , \text{其它} \end{cases} \end{cases} \quad (3.25)$$

1次B样条基函数为

$$\begin{aligned}
 N_{i,1}(u) &= \frac{W_0}{T_{10}} N_{i,0}(u) + \frac{-W_2}{T_{21}} N_{i+1,0}(u) \\
 &= \begin{cases} \frac{W_0}{T_{10}} & , u_i \leq u \leq u_{i+1} \\ \frac{-W_2}{T_{21}} & , u_{i+1} \leq u \leq u_{i+2} \\ 0 & , \text{其它} \end{cases}
 \end{aligned} \quad (3.26)$$

依次递推得3次B样条基函数为

$$\begin{aligned}
 N_{i,3}(u) &= \frac{W_0}{T_{30}} N_{i,2}(u) + \frac{-W_4}{T_{41}} N_{i+1,2}(u) \\
 &= \begin{cases} \frac{W_0^3}{T_{30}T_{20}T_{10}} & , u_i \leq u \leq u_{i+1} \\ \frac{-W_0^2W_2}{T_{30}T_{20}T_{21}} + \frac{-W_0W_3W_1}{T_{30}T_{31}T_{21}} + \frac{-W_4W_1^2}{T_{41}T_{31}T_{21}} & , u_{i+1} \leq u \leq u_{i+2} \\ \frac{W_0W_3^2}{T_{30}T_{31}T_{32}} + \frac{W_4W_1W_3}{T_{41}T_{31}T_{32}} + \frac{W_4^2W_2}{T_{41}T_{42}T_{32}} & , u_{i+2} \leq u \leq u_{i+3} \\ \frac{-W_4^3}{T_{41}T_{42}T_{43}} & , u_{i+3} \leq u \leq u_{i+4} \\ 0 & , \text{其它} \end{cases}
 \end{aligned} \quad (3.27)$$

由于权因子 w_i 、控制点 d_i 已知, 3次样条基函数可通过上式求得, 所以NURBS曲线上的位置向量即可求出。

3.2.3 轮廓误差分析与控制

NURBS曲线插补的轮廓误差主要来源于以进给步长为单位的短直线逼近实际曲线, 为了保证插补精度, 必须设定弓高误差, 使插补误差限制在给定的范围, 当误差超出了允许范围, 则按约束插补步长进行插补。NURBS曲线的误差分析可用下图3-7说明, ΔL 为插补进给步长, δ 为弓高误差, ρ 为曲率半径, 则它们之间的关系如下式(3.28)

$$\Delta L = 2\sqrt{\rho^2 - (\rho - \delta)^2} \approx \sqrt{8\delta\rho} \quad (3.28)$$

则弓高误差为

$$\delta = \frac{(\Delta L)^2}{8\rho} \quad (3.29)$$

约束插补步长为

$$\Delta L' = 2\sqrt{\rho^2 - (\rho - \delta_{\max})^2} \approx \sqrt{8\delta_{\max}\rho} \quad (3.30)$$

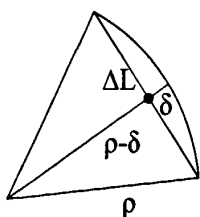


图 3-7 误差分析

Figure3-7 Error analysis

3.2.4 法向加速度分析与控制

由运动学知识我们知道，当物体沿着曲线运动时，由于曲率的存在，就会有沿曲线法向的拉力 F 。其计算公式如下式(3.31)，

$$F = ma \quad (3.31)$$

m 为刀具的质量， a 为刀具沿曲线的法向加速度， a 越大则 F 越大，若法向加速度过大，超过机床的承受能力将会影响零件的加工质量。因此，必须将其限制在允许的范围。

法向加速度的计算公式如下式(3.32)，其中 v 为切向速度， ρ 为曲率半径， a 跟 v 有关，因此可以通过限制切向速度来限制法向加速度。

$$a = \frac{v^2}{\rho} \quad (3.32)$$

3.3 速度控制

对于运动控制器来说，为避免电机在高速运转过程中因为频繁启停或频率发生突变而引起失步、超程和振荡等现象，保证运动的平稳性和准确的定位，必须对送到电机的进给脉冲频率或电压进行加减速控制^[17]。加减速控制是数控系统插补器的重要组成部分，是数控系统的关键技术之一，也是实现数控系统高实时性的瓶颈^[18]。

加减速控制可以在插补前后进行，在插补前进行叫前插补控制，在插补后进行

叫后插补控制。前加减速控制在插补前计算出进给速度，然后根据进给速度进行插补，得到各坐标轴的进给分量。所以它的优点是不影响实际插补输出的位置精度，但是它要预测减速点，计算量较大。后加减速的控制算法放在插补器之后，它的控制量是各动轴的速度分量。它不需要预测减速点，在插补输出为零时开始减速，并通过一定时间的延迟逐渐靠近程序段终点，其缺点是各坐标轴的合成位置可能不准确^[19, 20]。

常用的速度控制算法有梯形曲线加减速控制算法、指数加减速控制算法和S型曲线加减速控制算法等，下面分别对三种速度控制算法进行研究分析。

3.3.1 梯形曲线加减速控制算法

梯形曲线加减速控制算法是加减速控制方法中最为简单的一种，也是最常用的一种。它分为3个阶段：匀加速阶段，匀速阶段，匀减速阶段，其速度、加速度曲线如下图3-8所示。

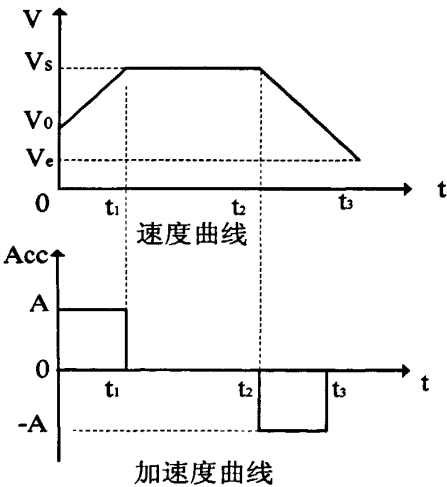


图 3-8 梯形加减速曲线图

Figure3-8 Trapezoidal Acceleration and Deceleration Curves

第一阶段为匀加速阶段(0~ t_1)，电机从起跳速度 V_0 ，以加速度 A 加速到稳定速度 V_s ；第二阶段为匀速阶段(t_1 ~ t_2)，电机以稳定速度 V_s 运行；第三阶段为匀减速阶段(t_2 ~ t_3)，电机从稳定速度 V_s ，以减速度 $-A$ 减速到 V_e 。

控制器每进行一次插补都要对稳定速度、瞬时速度、加减速进行处理，可用如下方法判断其加减速情况：

(1)加速处理

当程序给定的稳定速度大于当前的稳定速度便要进行加速处理，其瞬时速度的计算表达式为(3.33)，A为加速度，T为插补周期

$$V_{i+1} = V_i + AT \quad (3.33)$$

(2)减速处理

控制器每进行一次插补都要计算离终点的瞬时距离 S_i ，并与减速段的长度 S_d 比较，当 $S_i \leq S_d$ 时便进入减速段， S_d 的计算表达式如下式(3.34)， V_e 为线段终点速度， V_s 为线段的稳定速度

$$S_d = \frac{V_e^2 - V_s^2}{2A} \quad (3.34)$$

3.3.2 指数加减速控制算法

指数加减速控制算法一般具有三个阶段：加速阶段、匀速阶段和减速阶段，与梯型曲线加减速不同的是，在加/减速阶段速度随着时间按指数规律变化，其速度、加速度曲线如下图3-9所示，

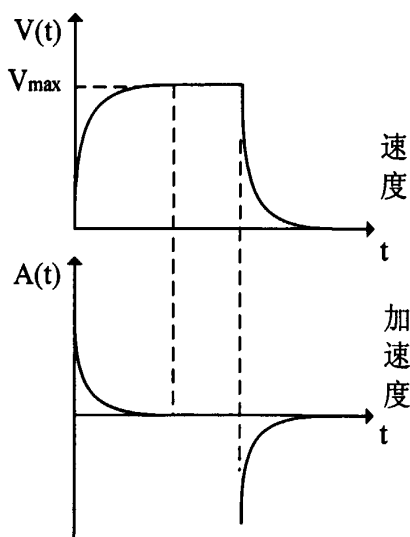


图 3-9 指数加减速曲线图

Figure3-9 Exponent Acceleration and Deceleration Curves

各阶段速度、加速度与时间的关系如下：

$$\text{加速段: } V(t) = V_{\max} (1 - e^{-t/T}), \quad A(t) = \frac{V_{\max}}{T} e^{-t/T}$$

$$\text{匀速段: } V(t) = V_{\max}, \quad A(t) = 0$$

$$\text{减速段: } V(t) = V_{\max} e^{-t/T}, \quad A(t) = -\frac{V_{\max}}{T} e^{-\frac{t}{T}}$$

其中 V_{\max} 、 T 分别为稳定速度和时间常数。

3.3.3 S曲线加减速控制算法

相对于梯形加减速控制算法，S曲线加减速控制算法柔性度较好，克服了电机在启动和加减速结束时存在的加速度突变^[20]。下图3-10是其位移、速度、加速度以及加加速度的运动学关系图，整个规划过程分七个阶段：加加速段、匀加速段、减加速段、匀速段、加减速段、匀减速段、减减速段。

假设电机的正向最大加速度和反向最大加速度相等，电机从0到最大加速度和从最大加速度到0所用的时间相等，如图中所示 $A_{\max}=A$ ， $A_{\min}=-A$ ， $T_1=T_3=T_5=T_7=T$ （ T 为系统的特性时间常数）， $J=A/T$ 。所以，加速度的计算公式为：

$$a(t) = a(t_i) + \int_{t_i}^t j(\tau) d\tau = \begin{cases} J\tau_1 & 0 \leq t \leq t_1 \\ A & t_1 \leq t \leq t_2 \\ A - J\tau_3 & t_2 \leq t \leq t_3 \\ 0 & t_3 \leq t \leq t_4 \\ -J\tau_5 & t_4 \leq t \leq t_5 \\ -J & t_5 \leq t \leq t_6 \\ -J + J\tau_7 & t_6 \leq t \leq t_7 \end{cases} \quad (3.35)$$

对加速度 $a(t)$ 积分得进给速度 $V(t)$ ：

$$V(t) = V(t_i) + \int_{t_i}^t a(\tau) d\tau = \begin{cases} V_0 + \frac{1}{2} J\tau_1^2 & 0 \leq t \leq t_1 \\ V_1 + A\tau_2 & t_1 \leq t \leq t_2, \quad V_1 = V_0 + \frac{1}{2} J\tau_1^2 \\ V_2 + A\tau_3 - \frac{1}{2} J\tau_3^2 & t_2 \leq t \leq t_3, \quad V_2 = V_1 + AT_2 \\ V_{\max} & t_3 \leq t \leq t_4, \quad V_{\max} = V_2 + AT_3 - \frac{1}{2} J\tau_3^2 \\ V_{\max} - \frac{1}{2} J\tau_5^2 & t_4 \leq t \leq t_5, \\ V_5 - J\tau_6 & t_5 \leq t \leq t_6, \quad V_5 = V_4 - \frac{1}{2} J\tau_5^2 \\ V_6 - J\tau_7 + \frac{1}{2} J\tau_7^2 & t_6 \leq t \leq t_7, \quad V_6 = V_4 - \frac{1}{2} J\tau_5^2 \end{cases} \quad (3.36)$$

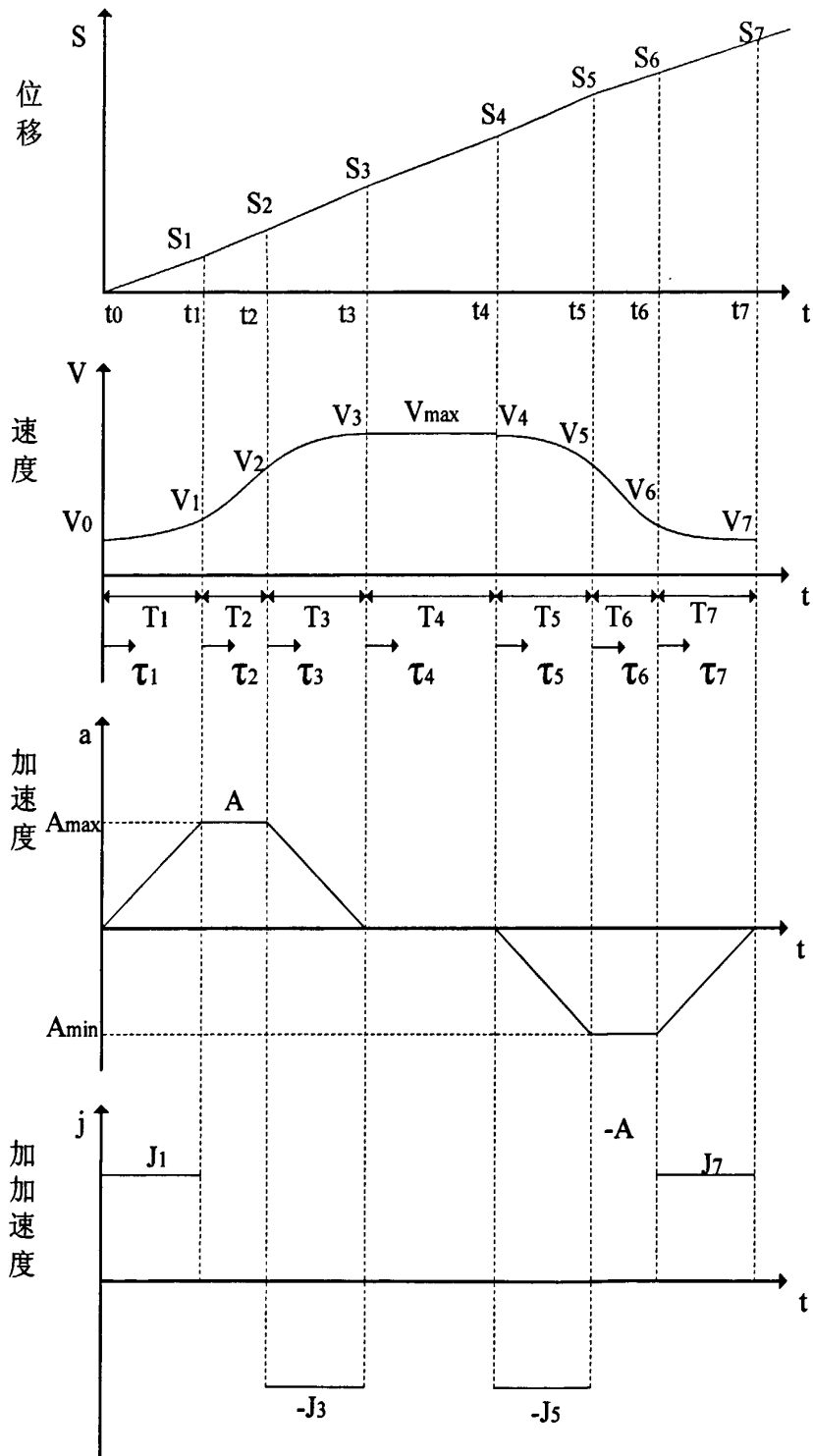


图 3-10 S 形加减速曲线图

Figure3-10 S-curve Acceleration and Deceleration Curves

对速度 $V(t)$ 积分得位移 $S(t)$:

$$S(t)=S(t_i)+\int V(\tau)d\tau=\begin{cases} V_0\tau_1+\frac{1}{6}J\tau_1^3 & 0\leq t\leq t_1 \\ S_1+V_1\tau_2+\frac{1}{2}JT_1^2 & t_1\leq t\leq t_2, S_1=V_0T_1+\frac{1}{6}JT_1^3 \\ S_2+V_2\tau_3+\frac{1}{2}JT_1^2-\frac{1}{6}J\tau_3^3 & t_2\leq t\leq t_3, S_2=S_1+V_1T_2+\frac{1}{2}JT_1T_2^2 \\ S_3+V_3\tau_4 & t_3\leq t\leq t_4, S_3=S_2+V_2T_1+\frac{1}{3}JT_1^3 \\ S_4+V_4\tau_5-\frac{1}{6}J\tau_5^3 & t_4\leq t\leq t_5, S_4=S_3+V_3T_4 \\ S_5+V_5\tau_6-\frac{1}{2}JT_5^2 & t_5\leq t\leq t_6, S_5=S_4+V_4T_5-\frac{1}{6}JT_5^3 \\ S_6+V_6\tau_7-\frac{1}{2}JT_5^2+\frac{1}{6}J\tau_7^3 & t_6\leq t\leq t_7, S_6=S_5+V_5T_6-\frac{1}{2}JT_5T_6^2 \end{cases} \quad (3.37)$$

由 $V(t)$ 可求得

$$\begin{cases} T_2 = \frac{1}{A} \left[V_{\max} - V_0 - \frac{1}{2}JT_1^2 - AT_3 + \frac{1}{2}JT_3^2 \right] \\ T_6 = \frac{1}{A} \left[V_{\max} - V_7 - \frac{1}{2}JT_5^2 - JT_7 + \frac{1}{2}JT_7^2 \right] \end{cases} \quad (3.38)$$

3.3.4 速度控制算法比较及选择

梯形加减速控制算法使电机在启动、停止或进给速度改变时,速度沿一定斜率的斜线上升或下降。这种算法运算简单、易于实现,占用CPU时间少,效率高,可于任意时刻改变运动装置的速度和位置。但是梯形加减速过程中加速度有突变,存在阶跃现象,使机床运动存在柔性冲击,速度的过渡也不够平滑,轨迹运动精度不是太高,所以此控制方法只适用于对加减速过程要求不高的场合^[22]。

指数加减速控制算法简单,适合于点位控制的加减速算法。与梯形加减速控制算法相比,速度平滑性较好,不易产生超调,运动精度较高,但在加减速的起点仍然存在加速度突变。

S曲线加减速控制算法在任意一点的加速度都是连续变化的,速度曲线平滑,运动精度高。相对于梯形加减速控制算法和指数加减速控制算法,S曲线加减速控制算法较复杂,计算量较大,多用于高速高精度加工中^[23, 24]。

通过以上比较分析可知,相对于梯形加减速控制算法和指数加减速控制算法,S曲线加减速控制算法消除了速度突变现象,加工质量要高,机械运行更平稳、噪音低,所以系统在加工过程中使用S曲线加减速控制。

3.4 本章小结

研究了运动控制中常用的数值插补算法(逐点比较法、数字积分法和时间分割法)、加减速控制算法(梯形加减速、指数加减速和S形加减速)及NURBS曲线插补的原理、特点。

第四章 系统硬件设计

本章将对系统的硬件设计展开分析，主要包括控制卡内部结构分析、运动轴行程限位保护、运动轴回零方法、硬件连接设计、抗干扰措施等。

4.1 控制卡工作原理及结构分析

课题采取“PC+运动控制卡”构成多轴运动控制器方案，其中的 6050 运动控制卡采用 DSP+FPGA 结构，其硬件结构图如下图 4-1 所示：

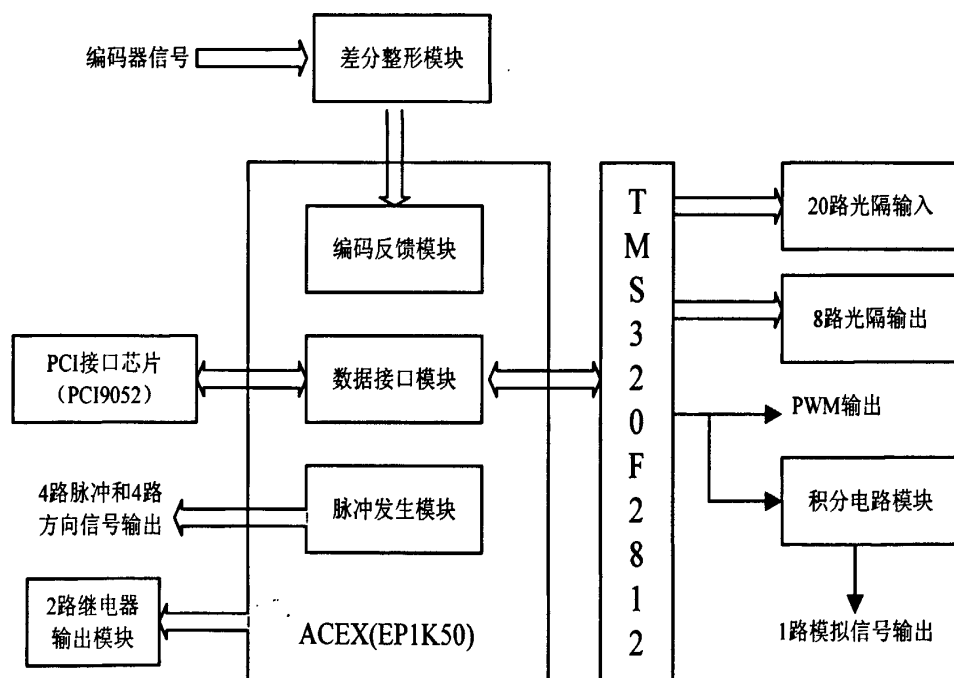


图 4-1 运动控制卡硬件结构图

Figure4-1 Motion Control Card Hardware Structure Diagram

运动控制卡硬件结构由与主机通信接口模块，DSP子系统，FPGA子系统，以及 I/O 接口电路四部分组成。通讯接口模块采用 PCI 总线方式完成与上位机通讯的硬件功能；DSP 子系统是该运动控制平台的计算核心，主要由 DSP、SDRAM、Flash 非易失存储器、时钟和电源管理等电路构成，负责管理整个系统的资源，运行多任务实时内核，并负担所有运动控制算法的计算任务；FPGA 子系统，实现运动控制的各种硬件功能，包括地址译码、逻辑控制、输入输出、脉冲序列发生、编码盘脉冲计数等；I/O 接口电路主要用于外围设备的控制及运动轴行程控制等。

控制卡软件工作原理框图如下图4-2所示，在FPGA中开辟有DSP与PC机的数据交换区，PC需要读取的数据（轴位置、速度、轴状态、插补状态及IO等），DSP会每0.5毫秒更新一次；而PC机在写入数据和命令时，先将数据写入数据交换区，然后向DSP发出中断信号通知DSP读取或执行命令。

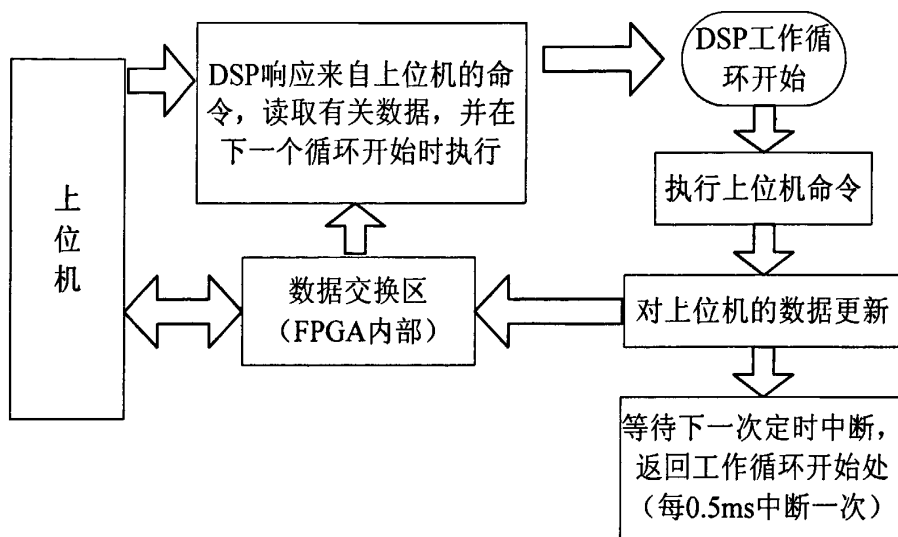


图 4-2 运动控制卡软件工作原理框图

Figure4-2 Motion Control Card Software Works Diagram

4.1.1 电源模块

6050运动控制卡由PC供电，并通过电源稳压芯片给板卡其它模块供电。主要模块有DSP模块、FPGA模块以及其它器件等。F2812芯片的CPU内核电压为1.8V，I/O电压为3.3V，并且F2812要求上电时I/O口先上电，内核后上电，掉电时I/O口先掉电，内核后掉电，所以在上电和掉电过程中需要有电源的时序配合。FPGA的内核电压为1.5V，I/O电压可以为3.3V、2.5V、1.8V、1.5V，但是其4个I/O BANK中的任意一个I/O BANK只能与其中的一种I/O电压相连。其它器件需要的电压一般为5V。

4.1.2 主控单元

主控单元主要完成复杂的运动控制算法，特别是位置控制和速度控制，插补迭代运算，开关输入和输出等实时要求高的任务^[25]，其性能决定着运动控制卡的速度和精度，运动控制卡的主控单元可以选择单片机、专用运动控制控制芯片或DSP。单片机的运算速度较低、片内资源不够丰富，难以完成高精度的运动控制要求；专

用运动控制芯片内部已固定了运动控制算法和所需的控制功能,具有集成度高、可靠性好等优点,开发人员只需按照其提供的使用方法加以调用即可,使开发周期大大缩短,但其价格昂贵,且不能改动内部算法,缺少灵活性;DSP具有多总线、多处理单元、流水线、硬件乘法器等结构,使其具有了高速数据处理和逻辑控制能力,能够较好地完成较为复杂的控制算法,而且许多芯片制造商已将电机控制所需的外围电路与DSP内核集成于一体,使得DSP成为高性能运动控制系统的首选器件。

6050运动控制卡的主控单元选用了TI的TMS320F2812,2812是32位定点DSP,主要使用在嵌入式控制应用,如数字电机控制、资料撷取及I/O控制等领域,其主要性能指标如下:

(1)高性能静态CMOS技术,主频最高可达150MHz,低功耗,1.8V内核电压和3.3V I/O电压供电;

(2)32位高性能CPU,可实现 32×32 位或双 16×16 乘法累加,哈佛总线结构,快速中断响应及处理外部事件,与TMS320F24x/LF240x DSP源代码兼容;

(3)片上存储器丰富,含有 $128K \times 16$ 位可加密FLASH, $18K \times 16$ 位SARAM,以及 $4K \times 16$ 位的引导ROM;

(4)外部接口高达 $1M \times 16$ 位的存储器,可编程等待周期,可编程的读/写选通时间,四个独立的片选信号;

(5)支持锁相环模块比率动态改变,具有片上振荡器及看门狗定时器模块;

(6)中断资源丰富,可支持45个外设级中断和3个外部中断,提取中断向量和保存现场只需9个时钟周期,响应迅速;

(7)片内功能丰富,两个事件管理模块,包括4路16位事件定时器,12路PWM输出通道及6路光电码盘接口;2路串行通信接口,1路SPI模式多通道缓冲串行接口和1路增强型局域网控制器模块;

(8)片上集成了16个通道的12位A/D接口,最小转换时间80ns,具有多达56个通用输入/输出多路复用器I/O口。

4.1.3 主机通信接口

运动控制卡需要与主机通信,接收上位机的指令及向主机反馈信息,完成复杂的运动控制,与主机的通信接口可选择ISA接口、USB接口或PCI接口。ISA是8-32总线,工作频率为8MHz,只能满足低速数据通信要求,而且现在的主板生产商趋向

于不再支持ISA, 所以现在的运动控制卡很少采用ISA接口; USB具有即插即用、支持热插拔、多级级联、传输距离远、带宽高、安装灵活等优点, 适合于需要经常插拔的计算机外设; PCI总线是Intel、Compaq等100多家成员支持的一种高性能、32 位地址/数据复用总线标准, 具有独立于处理器、高数据传速率、即插即用、低功耗、适应性强等特点, 是目前大多数运动控制卡所采用的主机通信接口。

目前实现PCI接口的有效方案有使用可编程逻辑器件和使用专用芯片两种。采用可编程逻辑器件实现PCI 接口的优点是能根据实际需要实现部分功能, 使设计运动控制卡所需的芯片减少, 集成度高, 抗干扰能力增强, 缺点是开发难度较高。采用专用接口器件的优点是能够有效的降低接口设计的难度, 缩短开发时间, 缺点是没有采用可编程逻辑器件那么灵活。

6050运动控制卡选用PCI总线接口芯片PCI9052与上位机之间实现通信, PCI9052是PLX公司开发的低价位PCI总线目标接口电路, 功耗低, 符合PCI2.1规范, 它的局部总线可以通过编程设置为8/16/32位的(非)复用总线, 数据传送率可达到132Mb/s, 其主要性能如下^[26]:

- (1)符合PCI总线2.1规范, 支持ISA适配器到PCI的转换;
- (2)支持局部总线到内存和I/O映射;
- (3)内含一个64字节的写FIFO和一个32字节的读FIFO, 从而支持预取模式的突发操作;
- (4)可通过EEPROM的配置改变局部总线的操作, 支持多路复用和非多路复用8位、16位和32位通用局部总线;
- (5)提供串行EEPROM接口, 从外部EEPROM中读取资源配置信息;
- (6)支持LOCAL BUS的等待状态。支持LOCALBUS读写操作的时延设置, 有内部等待时钟。

4.1.4 控制卡与PC机的通信模块

在开放式运动控制系统中, 运动控制卡与PC之间需要高速地传输大量的数据, 为了提高两者之间的数据传输速率, 一般采用双口RAM。双口RAM是特殊的RAM芯片, 它提供了两个完全独立的端口, 每个端口都有独立的地址、数据和控制信号, 允许两个端口独立地对存储器中的任何单元进行存取操作, 当两个端口同时对存储器中的同一单元进行存取操作时, 可由其内部仲裁逻辑决定优先权。在运动控制过

程中，DSP和PC机根据控制需要将其数据通过双口RAM的两个端口分别存放在双口RAM中，并读取对方的数据。

实现双口RAM的功能可以采用现成的双口RAM芯片或用FPGA来实现。采用双口RAM芯片可降低开发难度但增加了成本，而用FPGA来实现双口RAM的功能，可使得运动控制卡更集成化，体积更少。

4.1.5 通用数字I/O信号

1、输入口原理

6050控制卡有20个输入点，其中第9至第20点有双重目的，在缺省状态下，它们分别是4个轴的正限位输入点、零位输入点、负限位输入点。可用函数SetAxisIO_6050来控制它们是否有效，无效的轴IO可以当普通I/O来用。当轴I/O有效时，且当运动轴碰到限位开关时，输入点会以中断的方式通知DSP，DSP将以AbortAxis_6050模式使轴停止运动。I/O输入点还可接入急停开关，用Set_Emergency_Stop_6050函数可以使急停开关有效并与第1至第8某一输入点对应。缺省时急停开关无效。当急停信号到来时，DSP会以CeaseAxis_6050模式使全部轴停止。

6050控制卡的输入点使用DSP的I/O口读取，其具体过程是输入信号先经光耦隔离芯片TLP-521-4作信号隔离，采用光耦隔离可以有效地抑制尖峰脉冲和各种噪音干扰，然后经过处理后再由DSP读取外部输入信号，对于第9至第20输入点，由于其复用功能是4个轴的正负限位输入点和零位输入点，当其生效时需要DSP能及时响应，所以可用外部中断优先级来使DSP作出响应，其实现方式可将4个轴的正负限位输入点和零位输入点相与后接入DSP的外部中断，其实现过程可由FPGA模块实现。下图4-3是输入点原理图。

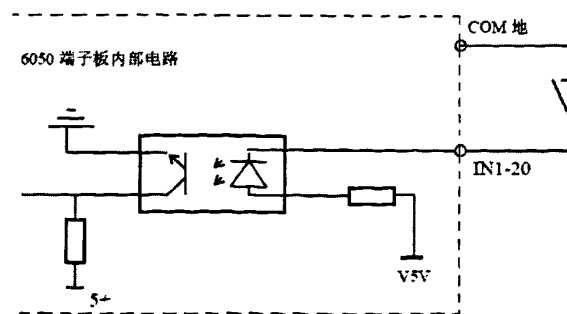


图 4-3 输入点原理图

Figure4-3 Input Schematic

2、输出口原理

6050 控制卡具有 8 路光隔/达林顿放大输出，最大驱动电流 500mA。其过程是 DSP 输出信号经过处理后输入光耦隔离芯片 TLP-521-4，由达林顿管驱动芯片 ULN2803APG 输出，二极管连接达林顿管集电极和电源正极的作用是防止感性负载在系统输出切断的瞬间产生过高的电压把三极管击穿。下图 4-4 是输出点原理图。

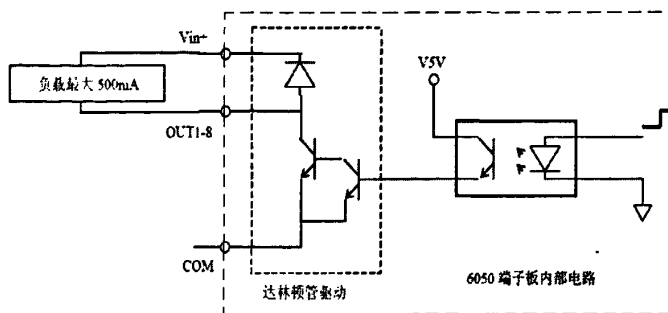


图 4-4 输出点原理图

Figure4-4 Output Schematic

4.1.6 PWM产生及模拟量输出

PWM信号由TI2812 DSP内部产生，其产生原理是：PWM的载波频率由DSP的主频（150MHz）进行1-128分频，形成PWM的基频，PWM发生器中有两个16位的计数器，这两个16位计数器再对基频进行分频，一个产生载波频率、一个产生占空比，所以6050运动控制卡所产生的PWM脉冲最低频率为：

$$150000000/(128*65536)=17.88\approx 18\text{Hz}.$$

此卡的模拟量输出没有使用到D/A转换芯片，而是将PWM转换为模拟量输出。在DA模拟量输出模式下PWM的载波频率固定为25KHz，所以此卡DA输出的分辨率为：1/6000，6000=150MHz/25KHz。

模拟量输出与PWM的关系可用下式(4.1)表示， τ 为PWM高电平的时间， T 为PWM的周期， V_o 为模拟量输出。

$$V_o = V_i \times \frac{\tau}{T} \quad (4.1)$$

PWM波形示意图如下图4-5所示，其转换为模拟量的方法可用下面的原理图4-6表示，PWM经光耦隔离后控制模拟开关的公共触点，使其与5V和地接通，在 V_i 得到与PWM一致的波形， V_i 经两级RC滤波后经电压跟随器输出，电压跟随器不仅可

以提高RC电路的带负载能力，而且使输出电压的线性度提高。

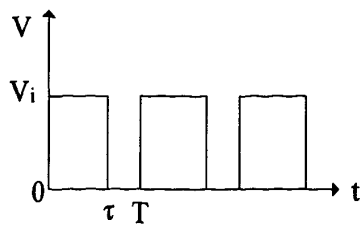


图 4-5 PWM 波形示意图

Figure4-5 PWM Waveform Diagram

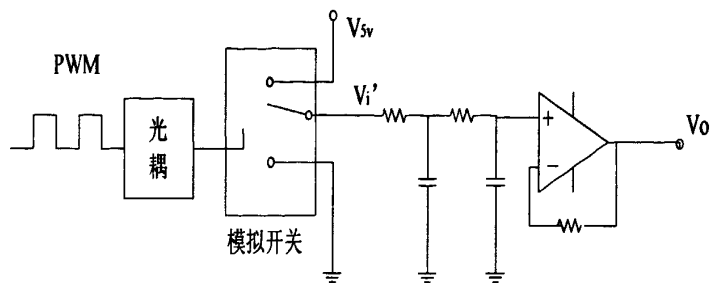


图 4-6 模拟量输出原理图

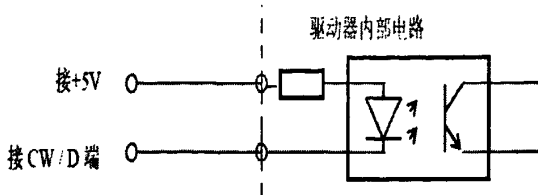
Figure4-6 Analog Output Schematic

4.1.7 电机控制模块

运动控制卡控制电机的方式有模拟量控制方式和脉冲控制方式两种。实现模拟量输出常用的方法有两种：一种是外接专用的DAC芯片，在高精度应用中，通常使用这种方法，但该方法成本较高；另一种可以通过微处理器的PWM输出滤波实现，这种方法精度稍差，但是资料分析可以达到12位精度，能够满足一般应用要求，6050运动控制卡有一路模拟量输出就是采用这种方法。

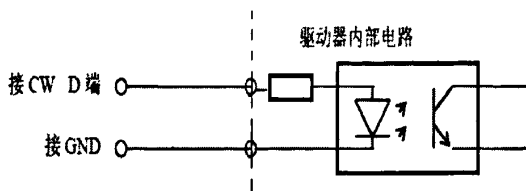
脉冲输出控制方式分为脉冲和方向、正转脉冲/反转脉冲、两相正交脉冲等三种，其实现方法通常是由DSP做粗插补，再由FPGA做精插补。由于差分信号对外部电磁干扰具有很强的抗干扰能力，尤其对共模干扰有很好的抑制作用，所以运动控制卡的输出接口电路通常会选用差分驱动器将单端信号转换为差分信号。

6050运动控制卡的电机控制方式采用脉冲输出控制方式，有脉冲和方向和正转脉冲/反转脉冲控制方式。其输出与驱动器的接线有共阳极接法和共阴极接法两种方式，分别如下图4-7和4-8所示：



4-7 共阳极接法

Figure4-7 Altogether anode connection



4-8 共阴极接法

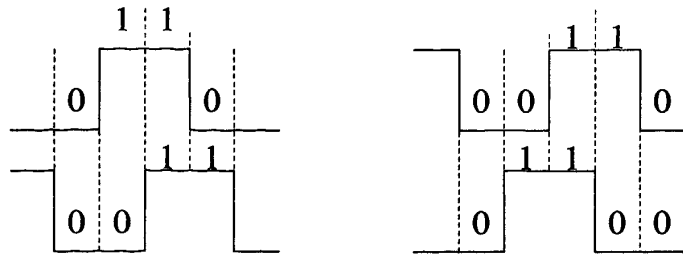
Figure4-8 Common cathode connection

4.1.8 编码器模块

检测元件是闭环伺服控制系统中必不可少的组成部分。在实际应用中，常采用脉冲编码器作为角位移传感器，把机械转角变成电脉冲。编码器分为光电式、接触式和电磁感应式。光电式脉冲编码器由于体积小、精度高、工作稳定可靠而应用广泛，光电脉冲编码器分为增量式脉冲编码器和绝对式脉冲编码器。绝对式脉冲编码器是通过读取编码盘上的图案来表示轴的位置，可以直接得到角度坐标的绝对值，其优点是没有累积误差，允许最高旋转速度高，缺点是结构复杂、价格昂贵；增量式编码器是直接利用光电转换原理输出三组差分方波信号，其优点是原理构造简单，机械平均寿命长，抗干扰能力强，可靠性高，适合于长距离传输，其缺点是无法输出轴转动的绝对位置信息^[27]。

6050运动控制卡的编码器模块是针对增量式编码器设计，其过程是使用差分芯片，将增量式编码器输出的三组差分方波信号转化为三组单极性的方波信号，然后输入FPGA的四倍频计数模块，计算出当前轴位置，然后被DSP读取。四倍频计数模块的作用是测量轴位置的精度，其原理是：当A相信号超前B相信号90°时，A、B两相信号的逻辑状态相对变化为“00→10→11→01”，当B相超前A相90°时，则为“00→01→11→10”，如下图4-9所示，只要将A、B两相信号的前一逻辑状态保存起来并在1/4周期内与当前逻辑状态进行比较，其变化情况如果符合“00→10→11→01”

或“00→01→11→10”中的一种，则计数器相应地执行加1或减1操作。



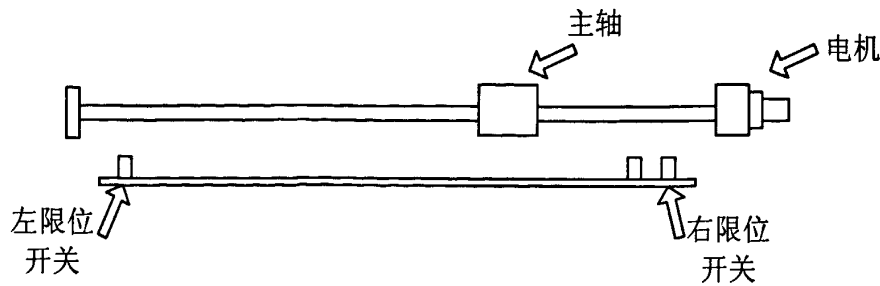
4-9 A、B 两相信号的时序逻辑图

Figure4-9 Timing Logic Diagram of A-B Phase Signals

4.2 行程限位保护

工作作进给运动时涉及到超行程限位保护的问题，每个轴的步进电机都必须在一定的安全范围之内运行，超过了这个安全范围就会造成工作平台的脱轨。超行程限位保护分为软超行程限位保护和硬超行程限位保护^[28]。软超行程限位保护，系统参数设置的每个运动轴移动极限位置是可以通过系统内部参数调整的；硬超行程限位保护，在运动轴行程极限位置的两端安装限位开关，当工作作进给运动碰撞到相应的限位开关，控制卡即时停止发送向这个方向运动的指令脉冲，从而起到运动限位保护的作用。

系统使用硬行程限位保护，在工作台的每个运动轴上装上接触式限位开关，并将其接入运动控制卡的限位输入口，当运动轴的机械模块碰到限位开关，限位开关即闭合，并以中断的方式通知DSP执行轴停止运动，下图4-10是限位开关安装示意图。

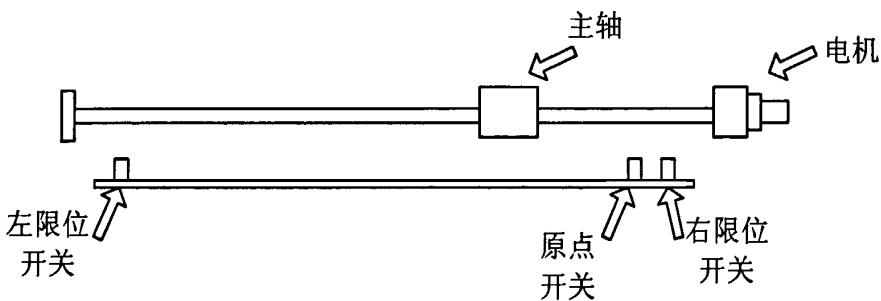


4-10 限位开关安装示意图

Figure4-10 Installation Diagram of Limit Switch0

4.3 运动轴回零

回零是数控系统中最重要的功能环节之一，是用以确定各坐标轴的位置。通过回零操作，确定各坐标轴零点，从而准确地建立物理坐标系。根据检测元件计量方式的不同又分为绝对栅格法回零和增量栅格法回零。绝对栅格法回零，系统只在首次开机调试时进行回零操作调整，同时系统后备存储器记录零点位置信息，此后开机不必再回零操作。增量栅格法，系统每次开机均必须进行回零操作^[29, 30]。下图4-11是原点开关安装示意图，将其装在每个运动轴的零点处，并接到控制卡零点信号输入端口。

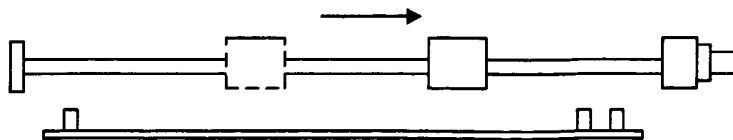


4-11 原点开关安装示意图

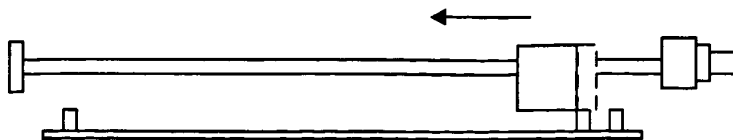
Figure4-11 Installation Diagram of Home Switch

6050运动控制卡的回零步骤如下：

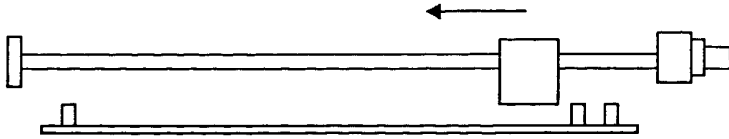
(1)运动控制卡收到回零命令，控制电机使运动轴以速度 V_1 从当前位置向零点方向移动。



(2)当运动轴碰到零点开关后，即以速度 V_2 反方向运动。



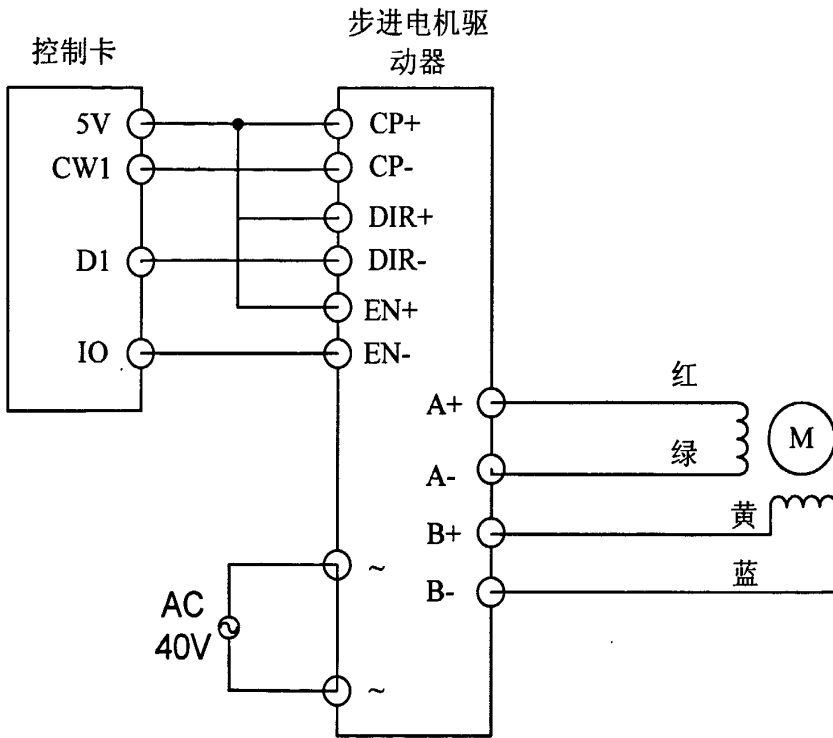
(3)当运动轴脱离零点开关一定距离后，运动控制卡使其立即停止运动，清零轴位置计数器，此时回零操作完成。



4.4 驱动器与电机连接

1、运动轴连接

下图 4-12 是控制卡其中一轴与步进电机及其驱动器的连接示意图，控制卡与



4-12 运动轴连接图

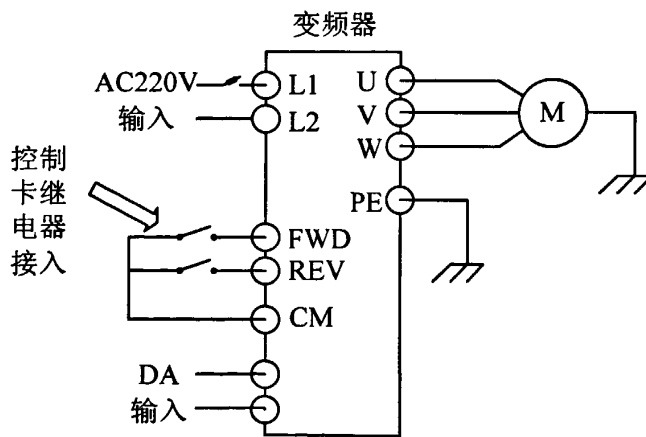
Figure4-12 Diagram of Movement Axis Connection

驱动器之间的控制信号有脉冲/方向和正向脉冲/反向脉冲两种模式，在此采用脉冲/方向模式控制步进电机驱动器，并采用共阳极接法，将控制卡的 5V 输出端口连接到驱动器的脉冲信号输入正端(CP+)和方向信号输入正端(DIR+)，控制卡的脉冲信号输出端(CW1)和方向信号输出端(D1)分别连接到驱动器的脉冲信号输入负端(CP-)和方向信号输入负端(DIR-)；EN+和 EN-为驱动器的使能信号输入端，当此信号回路导通过，驱动器停止工作，反则开始工作，在此本系统使用一个 IO 端出端子连接到 EN-，并使用共阳极接法对其控制，将控制卡的 5V 输出端口连接到 EN+；驱动器的电机接口输出信号 A+、A-、B+、B-分别与步进电机的红、绿、黄、蓝

相连；步进电机的步进角可通过驱动器的拨位开关(1-4)设定，共有 16 档细分，最小步进角为 0.009° ，每转 40000 步，最大步进角为 0.9° ，每转 400 步，在此本系统选择了步进角为 0.18° ，每转 2000 步，根据运动轴的螺距是 5mm，可以算出本系统的精度是 0.0025mm，理论上系统精度可达到 0.000125mm；驱动器的输出电流可通过拨位开关(7-10)设定，共 16 档可调，0-5A 范围，根据目前情况选择了 2A。

2、主轴连接

下图 4-13 是主轴连接示意图，变频器的运行命令通道有操作面板给定、控制端子给定和串行通讯口给定三种方式，考虑到接线和控制方便，本系统选择了控制端子给定方式，并通过变频器的控制面板进行相关设置。因此，主轴速度采用模拟量控制方式，模拟电压由控制卡给定，由于控制卡的电压输出范围是 $-10V \sim 10V$ ，变频器的电压输入范围有 0~5V 和 0~10V 两种，所以通过变频器的拨位开关选择其电压输入范围 0~10V，并在上位机软件中限制控制卡的电压输出在 0~10V，主轴运行频率是 400Hz，24000rpm 范围，通过对变频器相关参数设定，使控制卡 0~10V 的输出电压对应主轴 0~24000rpm；主轴的运转采用外部端子方式控制，将其连接到控制卡的继电器输出端口，FWD 和 CM 闭合，则主轴启动，否则停止，REV 和 CM 闭合，则主轴正向运行，否则反向运行；变频器的 U、V 和 W 输出端子分别与电主轴的相应端子相连，并将变频器和电主轴的外壳接地。



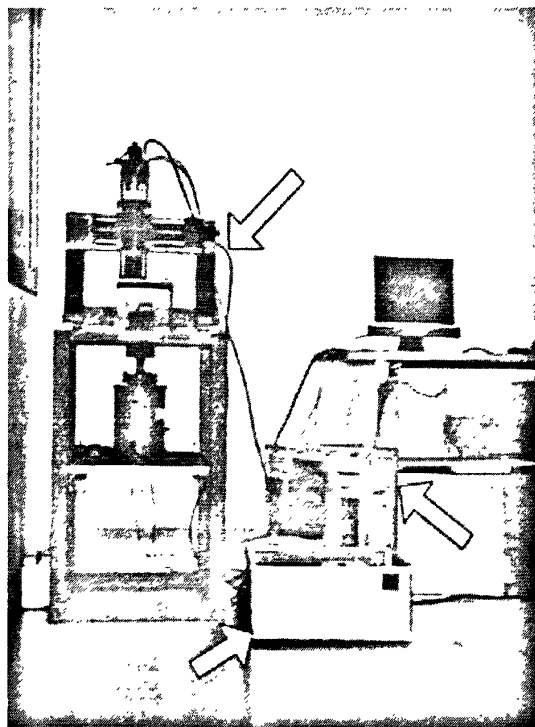
4-13 主轴连接图

Figure4-13 Diagram of Spindle Axis Connection

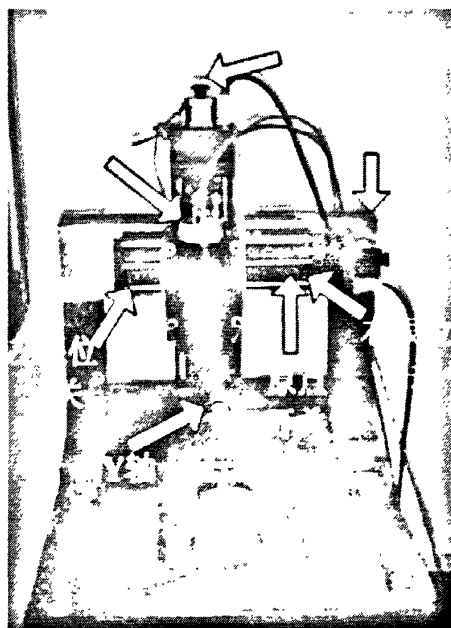
4.5 整体连接

每一个运动轴代表 3 维系统的一个方向，三个运动轴分别代表 X、Y、Z 方向。

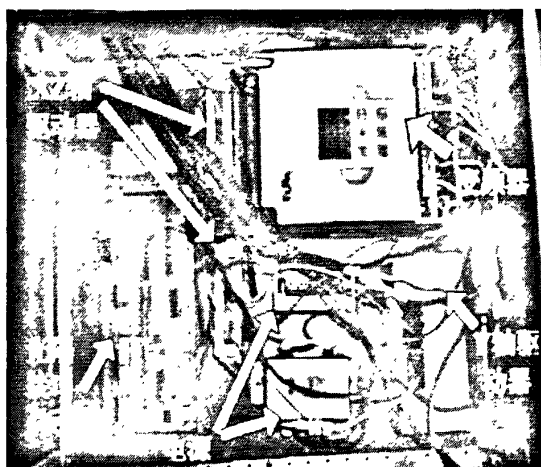
将限位开关、原点开关都安装在每个运动轴上，为了调试方便，将 I/O 控制板及驱动器都装在工控箱里，连接好的效果图如图 4-14 所示。



a) 系统整体



b) 加工平台



c) 工控箱

4-14 系统效果图

Figure4-14 System Effect Diagram

4.6 抗干扰措施

任何一个系统要应用到实际当中，其中最重要的一个原则是该系统具有高度稳定性，而系统不稳定通常是受到周围环境的干扰或设计缺陷等影响，控制系统受到干扰的表现通常有以下几种^[31, 32]：

- (1)控制系统没有正确响应上位机发出的各种指令。
- (2)光电编码器反馈回来的数值无规律乱跳。
- (3)与交流伺服系统共用同一电源的设备工作不正常。

系统使用的防干扰的措施主要有如下几种：

- (1)将电机的驱动器电源与控制系统的电源实行分组供电。
- (2)采用光电耦合器将控制系统与输入、输出通道隔离，光耦能有效地抑制尖峰脉冲及各种噪声干扰，提高信号传输过程中的信噪比。
- (3)将单端信号转换为差分信号，且采用双绞屏蔽线传输，这样能有效地抑制电磁信号的干扰。
- (4)将伺服系统及屏蔽线接地，防止静电感应和电磁干扰。

4.7 本章小结

主要对系统硬件的研究与构建，包括运动控制卡的内部硬件结构分析，控制卡与驱动器之间的连接方式分析，运动轴的限位及回零方法分析，系统的抗干扰措施分析等。

第五章 系统软件设计

本系统的上位机软件设计是基于图形化编程环境LabVIEW 8.6, 通过其CLF节点调用6050运动控制卡提供的Windows32动态连接库 (Dynamic Link Library, 简称DLL) 来编写与LabVIEW兼容的驱动程序, 实现该运动控制卡在LabVIEW环境下二次开发, 不仅缩短了开发周期, 而且使人机控制界面美观。

5.1 LabVIEW 简介

LabVIEW (Laboratory Virtual Instrument Engineering Workbench, 实验室虚拟仪器工程平台) 是美国NI公司推出的一种基于G语言 (Graphics Language, 图形化编程语言) 的虚拟仪器软件开发工具。用LabVIEW设计的虚拟仪器可脱离LabVIEW开发环境, 最终用户看见的是和实际的硬件仪器相似的操作面板。

LabVIEW是一个开放式的虚拟仪器开发系统应用软件, 它为设计者提供了一个便捷、轻松的设计环境, 利用它可以像搭积木一样, 轻松组建一个测量系统或数据采集系统, 并任意构造自己的仪器面板, 而无需进行任何繁琐的计算机程序代码的编写, 从而可以大大简化程序的设计。

LabVIEW与Visual C++, Visual Basic, LabWindows/CVI等编程语言不同, 后者采用的是基于文本语言的程序代码, 而LabVIEW则是使用图形化程序设计语言, 用方框图代替了传统的程序代码。LabVIEW所运用的设备图标与科学家、工程师们习惯的大部分图标基本一致, 这使得编程过程和思维过程非常相似。它广泛地被工业界、学术界和研究实验室所接受, 被视为一个标准的数据采集和仪器控制软件。利用它可以方便地建立自己的虚拟仪器, 其图形化的界面使得编程及使用过程都更加形象化^[33, 34]。LabVIEW具有以下特点^[35]:

(1)图形化的仪器编程环境: 它使用所见即所得的可视化技术建立人机界面针对测试、测量以及过程控制等领域, 提供了面板上所必需的许多显示和控制对象。

(2)灵活的程序调试手段: 用户可以在原代码中设置断点, 单步执行源代码, 在源代码数据流上设置断针, 在程序运行中观察数据流的变化。

(3)功能强大的函数库: LabVIEW提供了大量现成函数供用户直接调用, 包括基本的功能函数、仪器驱动程序和高级分析库(包括信号处理、窗函数、滤波器设计、

线性代数、概率论与数理统计、曲线拟合等)等,涵盖了仪器设计中几乎所有需要的函数。

(4)开放式的开发平台: LabVIEW提供了一些接口(如CLF接口和CIN接口)可与其它软件开发平台通信,构成更为强大的应用程序开发环境。

(5)网络功能: LabVIEW支持TCP/IP、DDE、IAC等功能。

5.2 LabVIEW 应用程序开发

5.2.1 LabVIEW应用程序构成

所有的LabVIEW应用程序,即虚拟仪器(VI),都包括前面板(Front Panel)、流程图(Block Diagram)以及图标/连接器(Icon/Connector)三部分^[96]。

1、前面板:前面板是图形用户界面,也就是VI的虚拟仪器面板,这一界面上有用户输入和显示输出两类对象(如开关、旋钮、图形以及其他控制和显示对象)。

2、流程图:流程图提供VI的图形化源程序。在流程图中对VI编程,以控制和操纵定义在前面板上的输入和输出功能。流程图中包括前面板上控件的连线端子、函数、结构和连线等。

3、图标/连接器:这部分的设计突出体现了虚拟仪器模块化程序设计思想。在设计大型自动检测系统时一步完成一个复杂系统的设计是相当有难度的。而在LabVIEW中提供的图标/连接工具正是为实现模块化设计而准备的。设计者可把一个复杂自动检测系统分为多个子系统,每一个都可完成一定的功能。

5.2.2 建立子VI

子VI(SubVI)相当于普通编程语言中的子程序,可以被其他VI调用,将子程序封装成子VI的形式可以实现模块化设计。因此在使用LabVIEW编程时,应与其他编程语言一样,尽量采用模块化编程的思想,有效地利用子VI,简化VI框图程序的结构,使其更加简单,易于理解,以提高VI的运行效率。建立子VI的步骤如下^[97]:

(1)创建连接端口:在完成一个VI的创建后,将其作为子VI调用的主要工作是定义VI的连接器。连接器是VI数据的输入输出接口,定义联接器的方法是先创建连接端口。用鼠标右键单击面板窗口中的图标窗口,在快捷菜单中选择“Show Connector”,此时,联接器图标会取代面板窗口右上角的图标。第一次打开连接端

口时, LabVIEW会自动根据前面板中的控制和指示对象建立相应个数的端口。此外, 用户可以根据自己的需要改变连接端口个数, 有两种方法, 一是在连接端口右键弹出选单中选择“添加接线端”或“删除接线端”, 二是在连接端口右键弹出选单中选择“模式”, 根据需要在下拉选单中选择相应的连接模式。

(2)定义连接端口关联关系: 完成了连接端口的创建后, 接下来就要定义面板中控制和指示对象与连接端口中各输入输出端口的关联关系。具体步骤是, 先在工具模板中选择连线工具, 然后单击前面板的一个控件, 此时控件图标的周围会出现一个虚线框, 再单击连接器图标上的一个连接端口, 这样, 控件就和此连接端口建立了关联关系。应用以上方法将其它控件和连接端口建立关联关系(一般控制对象的端子位于连接器窗口的左边, 显示对象的端子位于连接器窗口右边)。

(3)命名存盘: 完成步骤(1)和(2)后, 一个子VI已基本完成, 若子VI没有语法错误, 达到预期功能就可以将其命名存盘。

5.2.3 LabVIEW应用程序开发步骤

LabVIEW应用程序开发步骤如下^[10]:

(1)建立前面板: 从控制模板上选择所需的控件放在虚拟仪器的前面板上, 设置控件属性参数, 编辑控件标签, 设计布局前面板。

(2)建立流程图: 在流程图编辑窗口, 使用工具模板中相应工具, 根据程序要实现的功能, 从功能面板上选择并放置好所需对象(包括简单的数学运算、高级数据采集和分析方法、以及网络 and 文件输入输出操作等), 使用连线工具按数据流的方向将端口、节点、框图依次相连, 实现数据从源头按规定的运行方式送到目的终点。

(3)程序调试: 建立了前面板和框图程序后, 一个LabVIEW程序已基本建立, 程序中是否有语法错误, 可以通过工具栏上的运行按钮来辨别。若运行按钮是灰色的, 则表明框图程序有语法错误, 通过单击此按钮可查看程序的错误信息, 单击“显示错误”, 可在框图程序中高亮显示错误出处。此外, 可借助LabVIEW工具栏中的“高亮执行”、“单步执行”、“断点执行”等工具。

(4)运行检验: 编写的LabVIEW程序是否达到预期功能, 还需运行检验, 可通过以下两种方式检验。

仿真检验: 不使用I/O接口硬件设备, 对VI检验运行所需的信号数据采用由数组或信号发生函数生成的仿真信号。

实测检验：通过I/O接口硬件设备来检验虚拟仪器的功能。

(5)命名存盘：反复通过以上步骤(2)、(3)和(4)使程序达到预期的功能，然后将LabVIEW程序命名存盘。

5.3 控制卡基于 LabVIEW 驱动研究

LabVIEW具有强大的外部接口能力，可用的外部接口包括：DDE、CIN、MATLAB Script、CLF等，其中通过CLF调用DLL是其常用的外部接口技术^[39, 40]。

DDE即动态数据交换，是Windows操作系统中一种基于消息的协议，它是运用内存共享技术，以客户应用程序和服务器应用程序成对出现的形式来实现信息交换的。利用DDE，LabVIEW可以与Windows环境下的多种软件开发平台，如VB、VC、Excel等发送或接收数据。

CIN即代码接口点，是一个能与用传统语言(C、Pascal、Basic)编写的可执行程序相关连，带有输入输出端口的图形化节点。利用CIN技术，LabVIEW可以对其不擅长或不易实现的任务通过适于完成此类任务的其它编程语言来实现，从而提高程序执行效率。

MATLAB Script节点是LabVIEW与MATLAB数据通信的方法。由于LabVIEW不擅长于大量的、复杂的数学运算，而Matlab是一种常用的高效率数学运算工具，是建立在向量、数组和复数矩阵的基础上，使用方便，利用此节点可以将两者有机地结合起来，大大减少编程的工作量，提高编程效率。

系统使用的运动控制卡非NI公司的产品，不带LabVIEW驱动程序，但其提供了DLL形式的I/O驱动程序，所以本文利用LabVIEW与外部编程语言之间的接口技术CLF来间接调用其基于C语言编写的驱动程序。

5.3.1 动态链接库简介

动态链接库(Dynamic Link Library, 简称DLL)是一个包含可由多个程序同时使用的代码和数据的库，是基于Windows程序设计的一个非常重要的组成部分。相对于静态链接库，动态链接库所调用的函数并没有被拷贝到应用程序的可执行文件中，而是在应用程序运行期间被动态连接。动态链接库可以和其它应用程序共享库中的函数和资源，因此有助于减小内存的开销，同时提高代码的重用率，便于实现混合编程。

5.3.2 调用库函数节点技术

LabVIEW提供了调用库函数节点(Call Library Function node, 简称CLF), 利用CLF, 可以在LabVIEW中实现DLL和API函数的调用。在LabVIEW中使用CLF需要了解动态链接库里被调用的函数名称、功能及其输入输出参数等, 下面将详细介绍在LabVIEW 8.6下利用CLF调用6050运动控制卡基于C语言编写的驱动程序^[4]。首先在框图程序中创建一个空的CLF, CLF位于“函数模板→互连接口→库与可执行程序子模板”, 如图5-1所示, 下面给出其详细的配置过程:

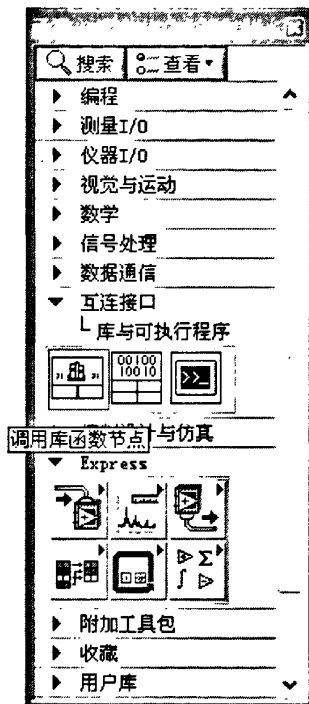


图 5-1 CLF 节点图标

Figure 5-1 CLF Node Icon

(1) 双击CLF图标弹出配置对话框, 如图5-2所示, 首先要指定6050控制卡动态链接库的库名或路径。点击其右边的文件夹图标, 打开一个文件对话框, 找到6050运动控制卡的动态链接库文件“dfjzh6050dll.dll”所在位置。

(2) LabVIEW装载了DLL文件后会自动检测其所包含的函数, 但不能检测函数中的参数及其数据类型, 因此接下来的要参照DLL说明文件“dfjzh6050dll.h”及控制卡使用手册选择所要配置的函数名, 设置函数参数。

(3) 在调用规范中有“stdcall(WINAPI)”和“C”两个选项, 它们的区别是, 若调用函数为Windows标准共享库函数则选择“stdcall(WINAPI)”选项, 若调用函数

为VC环境下编译的库函数则选择“C”选项，本文选择“C”选项。

(4)在“线程”选项中选择“在UI线程中运行”。

至此，完成了所有配置选项，在CLF配置图最下方可以看到函数原型。下面阐述配置CLF过程中需注意的两个问题：

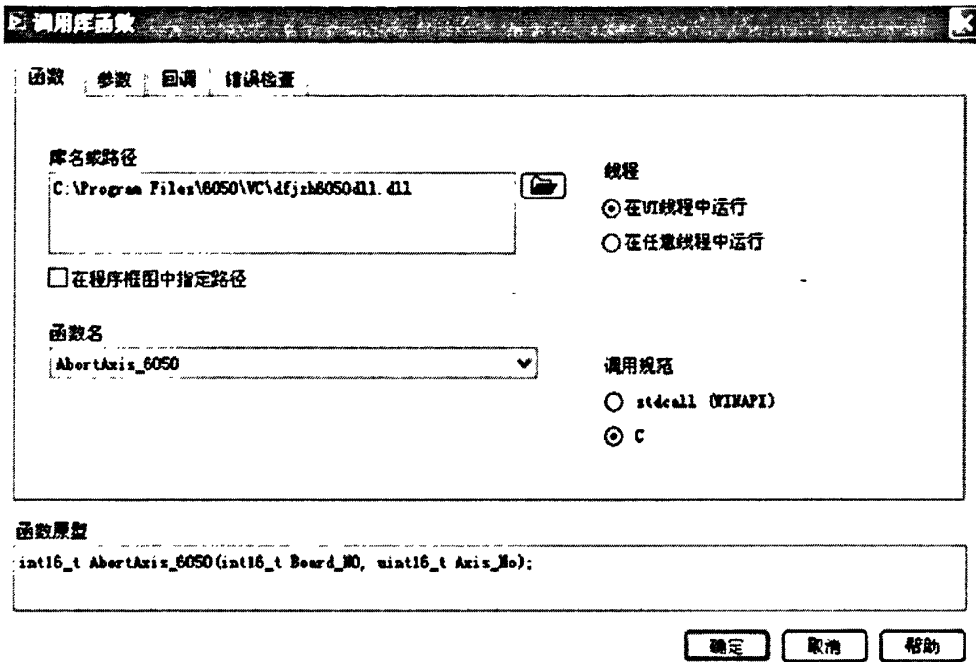


图 5-2 CLF 节点配置图

Figure5-2 CLF Node Setting Chart

(1) LabVIEW线程调用方式

节点配置中，有两种线程调用方式，“在UI线程中运行”和“在任意线程中运行”。“在UI线程中运行”表示在用户接口线程中调用，DLL的执行期将等到用户接口线程（在此即LabVIEW环境下的VI应用程序）执行DLL的导出函数调用时才开始；“在任意线程中运行”表示允许多个线程同时调用这个DLL。当DLL仅由一个线程调用，两种调用方式都可选择，若出现多个线程调用情况且不确定DLL是否为线程安全，选“在UI线程中运行”。

(2)函数参数的配置

对于所调用的DLL库函数的参数类型，如果在配置框中找不到匹配的，可以在类型下拉列表中选择“匹配至类型”，在数据格式下拉列表中选择“按值处理”，表示将编程时指定的LabVIEW数据类型与该参数数据类型自动匹配。下表5-1为常见的LabVIEW和32位系统C语言数据类型对照表^[42]

表5-1 两种变量类型的对应关系

Table5-1 Correspondence between the two Types of Variables

	VC++数据类型	LabVIEW对应的数据类型	备注
简单基本类型	Char	Int8	对于这些基本类型的指针的指针类型变量，应在Parameters栏中的“Pass”传递类型中选择“Pointer to value”(通过指针传递)而不是“Value”(值传递)
	Unsigned char	Uint 8	
	Short int	Int 16	
	Int	Int 32	
	Long	Int 32	
	Unsigned long	Uint 32	
	Float	Float 32,4 B single	
	Double	Float 64,8 B double	
数值数组类型	一维数值型数组	Array	具体设置：Type→Array，DataType→对应数组的基类型，Dimensions→对应数组的维数，ArrayFormat→Array Data Pointer
	二维数据值型数据	Array	
	指向数组的指针	没有对应类型	如果不使用LabVIEW本身提供率，而单纯使用VC++，则不能在LabVIEW中正确使用
	指向指针的数组	没有对应类型	
字符串	Char *	string	具体设置：Type→string，string Format→C string pointer
	Char[n]		
	Char[][]	没有对应类型	如果不使用LabVIEW本身提供率，而单纯使用VC++，则不能在LabVIEW中正确使用
	Char**		

结 构 体	sPeople tmp	没有对应类型	设结构体名称为：sPople， 且结构体内变量定义不能 使用LabVIEW中不支持的 类型。在进行连线时，要根 据结构内变量来定义Cluster
	sPeople* tmp	可以使用Adapt to Type	具体设置：Type→Adapt to Type,Data Format→Handles by value

考虑到程序的模块化设计和控制卡基于LabVIEW的驱动程序重复调用的问题，参照5.2节建立子VI的方法，基于CLF技术将动态链接库中所有的函数封装为子VI的形式，如下图5-3所示，LabVIEW通过调用下图子VI即可实现与6050运动控制卡通信，完成运动控制功能。

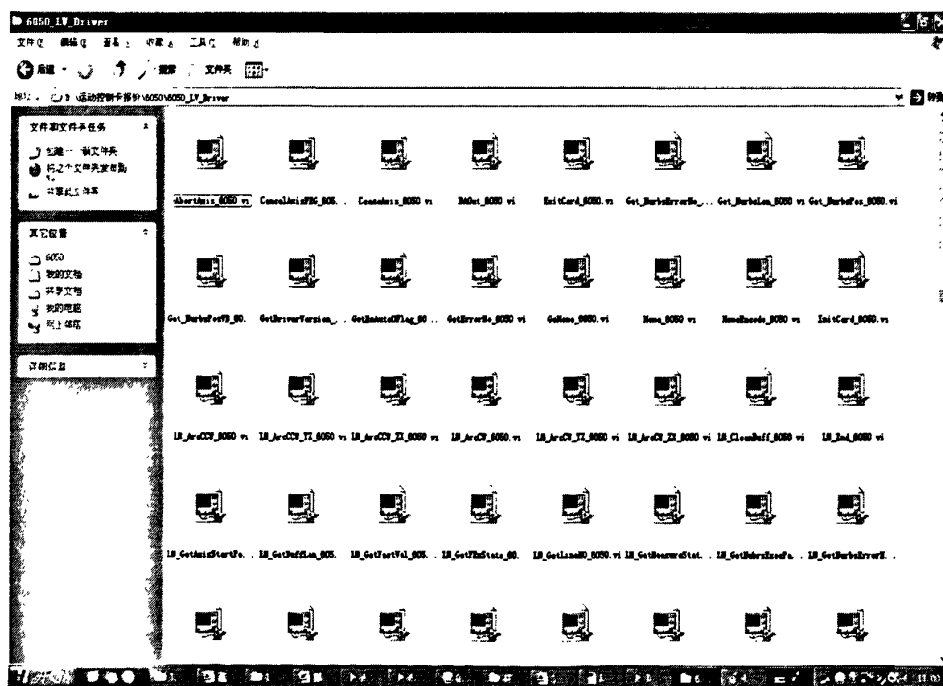


图 5-3 函数库

Figure5-3 Diagram of Library

5.4 人机控制界面

系统人机控制界面如下图5-4所示，主要包括四部分：运动轨迹显示部分、运动轨迹参数设置部分、PWM/DA输出部分和I/O控制部分。

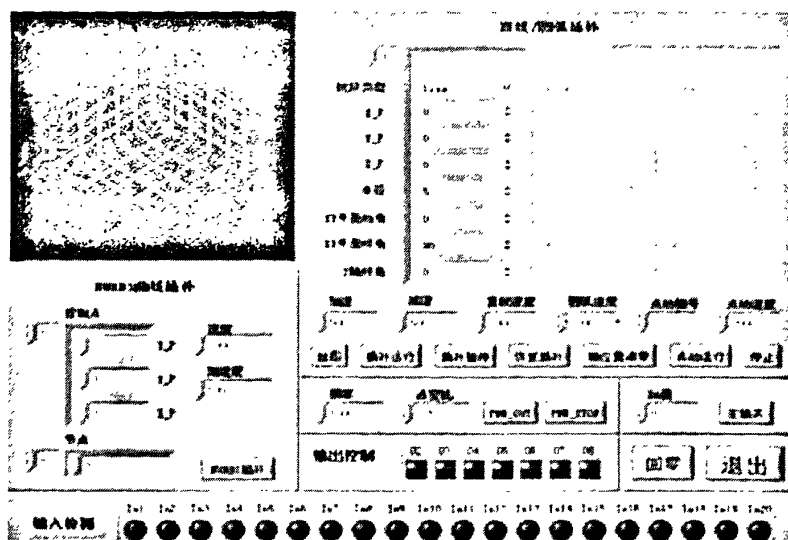


图 5-4 人机控制界面

Figure5-4 Man-machine Control Interface

运动轨迹显示部分利用“3D Curve”控件，“3D Curve”控件可用于显示三维曲线，其在流程图中的图标如下图5-5所示，它由两部分组成，一个是ActiveX控件3D Curve图标，另一个是3D Curve.vi图标，由其属性可知，该控件只能以向量的形式输入X、Y、Z三轴的坐标，否则会出错；对于每一线段参数的设置，程序采用数组与簇相结合的方式将其存放。

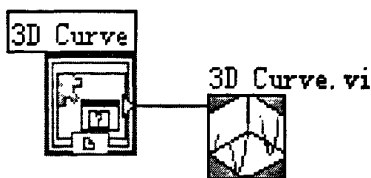


图 5-5 3D Curve Graph 图标

Figure5-5 3D Curve Graph Icon

对于界面中每一按钮的执行动作，程序使用LabVIEW的“事件结构”来响应，事件结构的工作原理和内置的条件结构相似，可包含多个分支，每一个分支对应一个独立的事件处理程序。一个分支配置可处理一个或多个事件，但每次只能发生这些事件中的一个事件。事件结构执行时，将等待一个之前指定事件的发生，待该事件发生后即执行事件相应的条件分支，一个事件分支处理完毕后，事件结构也就执行完毕^[7]。事件结构由事件选择标签，事件数据节点，时间接线端和程序框构成，如图5-6所示，需要注意的是事件结构必须放在While循环里，否则，无论有多少事

件发生只能对第一个发生的事件进行处理，处理完后程序将退出事件结构。

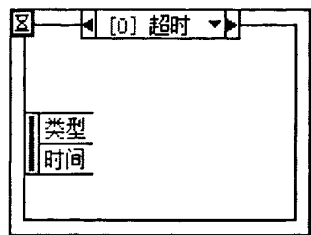


图 5-6 事件结构图标

Figure5-6 Event Structure Icon

5.5 程序实现

系统程序主要分为两大部分，即仿真程序和控制程序。其中仿真程序包括直线仿真程序和圆弧/螺旋线仿真程序以及总的仿真程序实现方法；控制程序包括三维运动轨迹控制、PWM/DA输出和I/O控制等。

5.5.1 仿真程序实现

1、直线仿真子程序

对于直线，我们可以将其起点和终点坐标用创建数组函数连接起来，但要分两种情况，一是起点是原点，另一种情况是起点是前一线段的终点。对于前者，要将原点和终点的三轴坐标分别用创建数组函数连接，对于后者，只需将其终点的三轴坐标分别输入创建数组函数，“3D Curve”控件会自动将前一线段的终点和此点连接。下图5-7是直线仿真子程序截图：

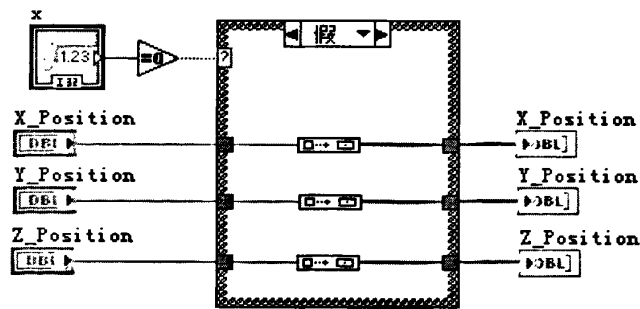


图 5-7 直线仿真子程序

Figure5-7 Linear Simulation Subroutine

其起点是前一线段的终点，输入变量X_Position、Y_Position和Z_Position是直线

段的终点在三坐标轴的分量，输入变量X是用于判断起点是否为原点，其原理是：由于while循环的i变量是从零开始的，每经过一次循环i变量增加1，所以可将其接入变量X，当i为零时起点即为原点。三个输出变量是以数组形式输出的三轴坐标分量。将直线仿真子程序封装成子VI的形式如下图5-8所示。

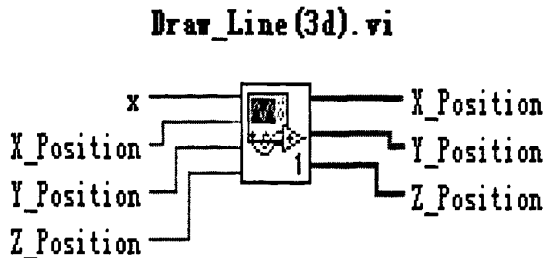


图 5-8 直线仿真子程序子 VI

Figure5-8 Sub-VI of Linear Simulation Subroutine

2、圆弧/螺旋线仿真子程序

与直线仿真子程序不一样，要使仿真出来的圆弧/螺旋线圆滑，需对其采集若干个数据点，然后将N个数据点的三轴坐标向量用LabVIEW的创建数组函数连接起来，再输入“3D Curve”控件，以下图5-9螺旋线为例说明圆弧/螺旋线仿真的实现。由图可见，圆弧SA的半径为R， $\angle SOA = \alpha$ ， $\angle AOE = \beta$ 。当 $\angle \beta = 0$ 时，此曲线即为圆弧SA，当 $\angle \beta$ 不为0时，此曲线即为螺旋线SE。

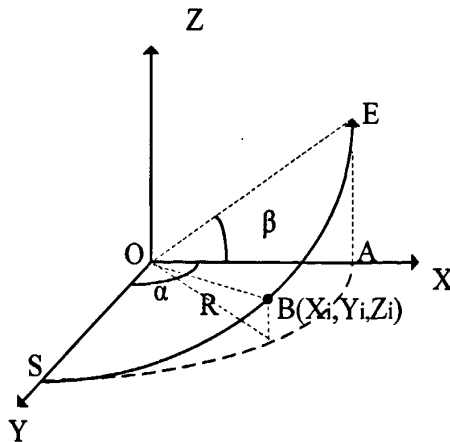


图 5-9 螺旋线

Figure5-9 Spiral Diagram

沿着圆弧SA，每移动 0.1° 对螺旋线SE采集一个数据点，则可以对其采集 $N = 10\alpha$ 个数据点，对于螺旋线SE上的第n个数据点 $B(X_i, Y_i, Z_i)$ ，其坐标可用下式(5.1)计算

$$\begin{cases} X_i = R \times \cos(\angle OSY + n \times \frac{\alpha}{N}) \\ Y_i = R \times \sin(\angle OSY + n \times \frac{\alpha}{N}) \\ Z_i = R \times \tan(n \times \frac{\beta}{N}) \end{cases} \quad (5.1)$$

式中 $\angle OSY$ 为螺旋线SE在XY平面的起始角度，用LabVIEW程序实现此数据点的采集如下图5-10所示。输入变量Yaw_S_Angle、Yaw_T_Angle、Pitch_T_Angle、Radius、X_center和Y_center分别是螺旋线在XY平面投影的起始角度、终止角度、螺旋线终点沿Z轴的角度、螺旋线在XY平面投影的半径和圆心，程序通过“for loop”计算出螺旋线每一采样点在X、Y和Z轴的坐标，然后以向量的形式输出三轴坐标。

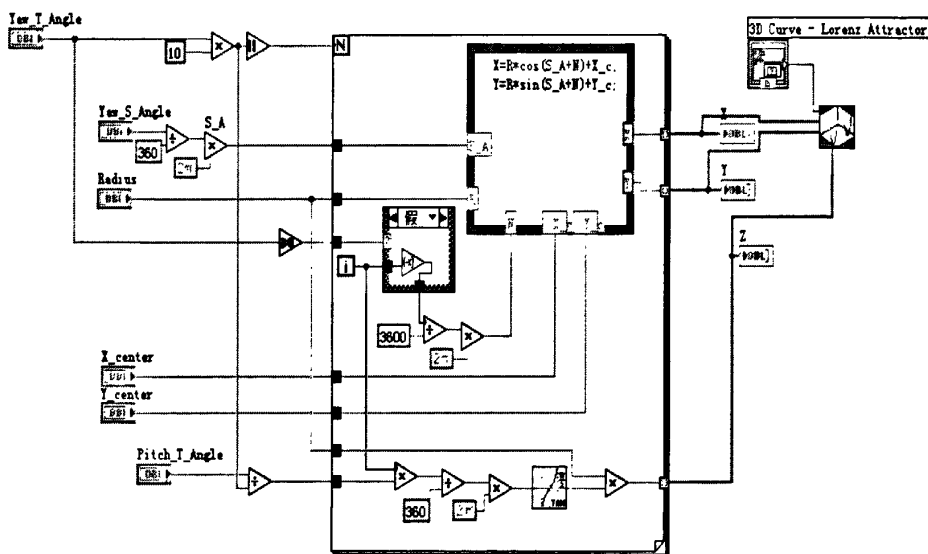


图 5-10 螺旋线数据点采集程序

Figure5-10 Spiral Data Point Collection Program

程序中用到公式节点，此公式节点里的数学公式和和C语言相似，对于较为复杂的数学公式用C语言去实现比用LabVIEW的G语言更为简洁、方便，此乃LabVIEW的一大特色之一。公式节点的图标如下图5-11所示，它的使用方法是：

(1)创建输入输出端口：先在其框架的右键弹出选单中选择“添加输入”，然后在出现的端口图标中填入该端口的名称，就完成了输入端口的创建，输出端口的创建与此类似。值得注意的是所有输入变量的端口都在公式节点框架的左边，输出变量的端口都在框架的右边。

(2)输入程序代码：按照C语言的语法规则在公式节点的框架中加入程序代码，

与C语言不同的是公式节点框架内的每个公式后都必须有分号标示。

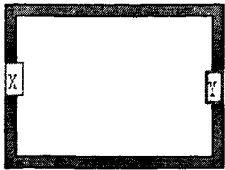


图 5-11 公式节点图标

Figure5-11 Formula Node Icon

将螺旋线仿真子程序封装为子VI的形式如下5-12所示：

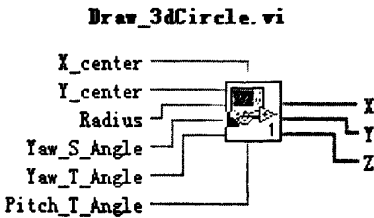


图 5-12 螺旋线仿真子程序子 VI

Figure5-12 Sub-VI of Spiral Simulation Subroutine

3、仿真程序实现

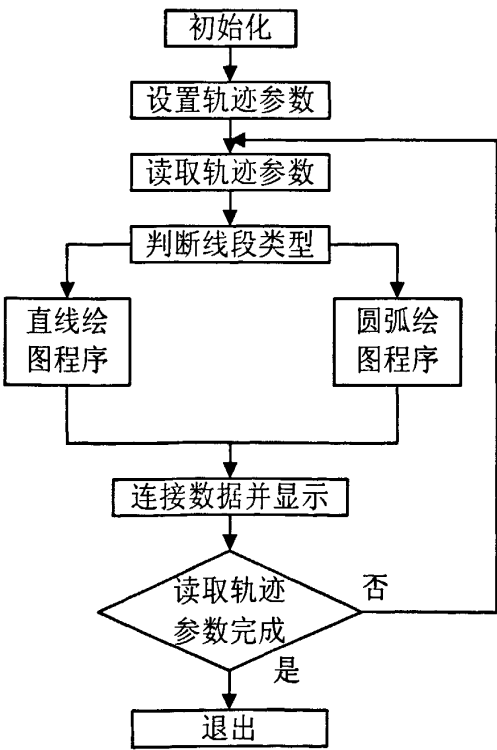


图 5-13 仿真程序流程图

Figure5-13 Simulation Program Flow Chart

移位寄存器的功能是把当前循环完成时的某个数据传递给下一次循环的开始，它在流程图上用在循环边框上相应的一对端子来表示。右边的端子中存储了一个周期完成后的数据，这些数据在这个周期完成之后将被转移到左边的端子，赋给下一个周期。移位寄存器可以转移各种类型的数据——数值、布尔数、数组、字符串等，它会自动适应与它连接的第一个对象的数据类型。

程序使用了“移位寄存器”将当前数据保存起来并送入下一次循环，并用创建数组函数将前一段的数据与当前线段数据连接起来，这样在while循环里就可以将所有的线段连接起来并在“3D Curve Graph”控件中完整地显示整条运动轨迹。

5.5.2 控制程序实现

1、直线、圆弧控制子程序

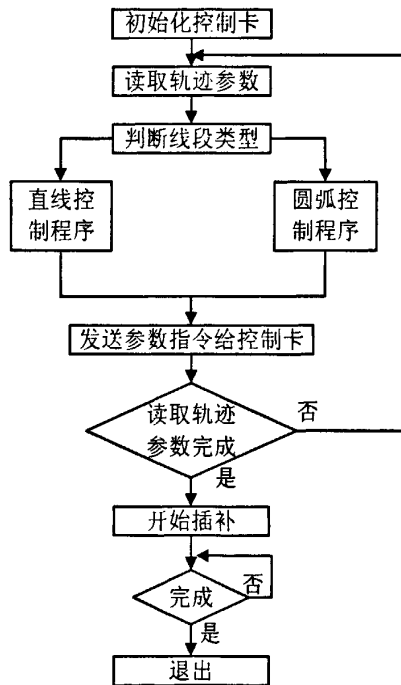


图5-15 控制程序流程图

Figure5-15 Control Program Flow Chart

程序利用LabVIEW的CLF技术调用运动控制卡的动态链接库函数，将用户设置的运动轨迹参数发送给运动控制卡，由运动控制卡去控制步进电机实现用户预设的运动轨迹，其程序流程图如上图5-15所示，先读取轨迹参数，判断其线段类型，然后进入相应的子程序计算参数，再发送到控制卡，当读取所有的线段完成后开始执行插补命令，否则继续读取线段参数，插补完成后退出程序否则继续插补。

轨迹控制的核心程序如下图5-16所示，该程序通过自动索引隧道读取每一线段的参数，然后通过条件结构判断该线段是直线或圆弧，然后进入相应的程序，将输入参数计算，转化为运动控制卡控制函数所需的参数，再通过CLF调用运动控制卡的驱动发送给运动控制卡^[43]。

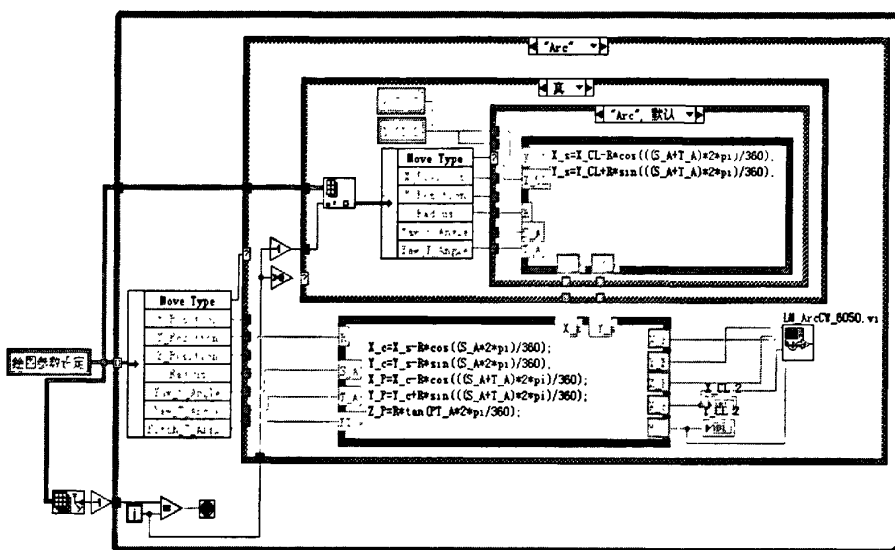


图 5-16 轨迹控制核心程序

Figure5-16 Core Program of Trajectory Control

对于圆弧/螺旋线，其插补命令函数原型为“short LM_ArcCW_6050(short Board_NO, double xPos,double yPos,double zPos,double wPos,double xcPos,double ycPos,double Speed,long LineNO)”，所以需计算出其圆心位置，以下图 5-17 第一象限圆弧为例说明其计算方法，

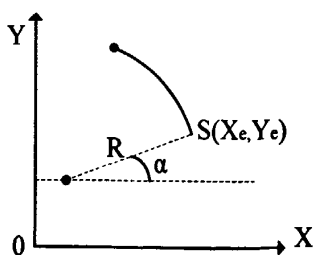


图 5-17 计算圆弧圆心

Figure5-17 Diagram of Calculating the Center of Arc

前一线段的终点是 $S(X_e, Y_e)$, 当前圆弧的半径为 R , 起始角度为 α , 则圆弧的圆心为

$$\begin{cases} X_c = X_e - R \cos \alpha \\ Y_c = Y_e - R \sin \alpha \end{cases}$$

2、NURBS曲线插补控制程序

下图5-18是NURBS曲线插补控制核心程序，控制点参数至少为4个，且节点矢量数 $m=n+k+1$ （ n 为控制点数， k 为曲线指数），否则认为输入参数错误，程序不执行，权因子设为1。

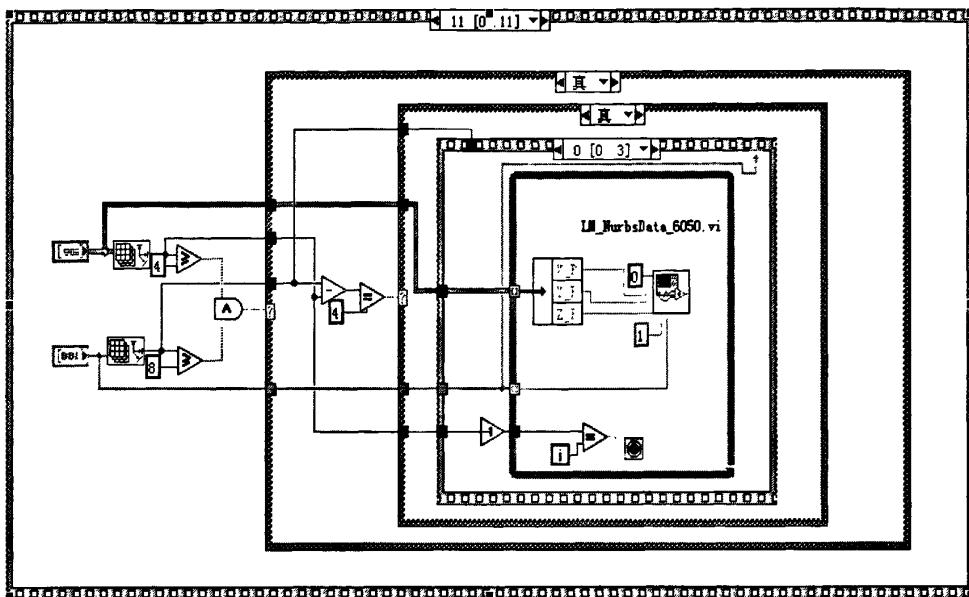


图 5-18 NURBS 曲线插补控制程序

Figure5-18 Diagram of NURBS Curve interpolation control procedure

3、主轴控制

主轴速度采用模拟量控制，电压输出范围0~10V对应着变频器0~400Hz，通过DAOut_6050()函数设定电压输出值。主轴的启停通过函数WriteIoBit_6050()控制继电器的断开或闭合。

在本系统中，PWM和DA输出使用的是同一个端子，通过跳线设置其输出方式。PWM输出是本系统的扩展功能，在实际应用中，PWM 输出可控制激光电源。

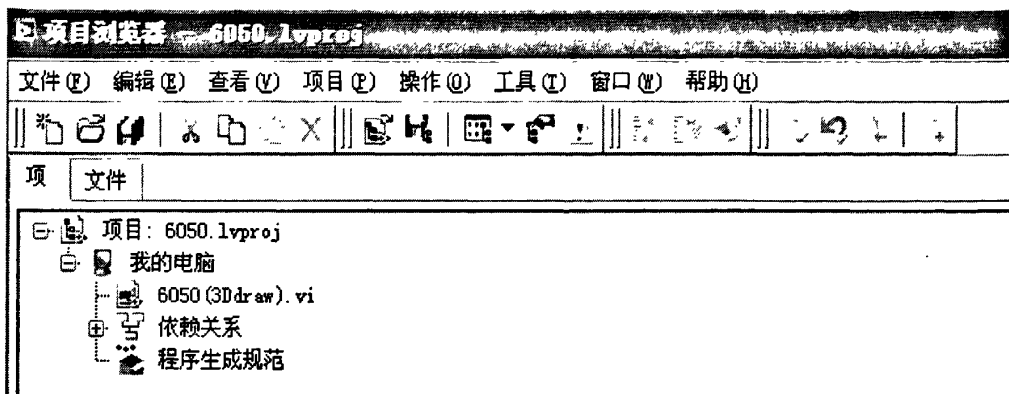
4、I/O控制

I/O控制也是本系统的扩展功能，可通过CLF节点调用ReadIO_6050和WriteIo_6050函数实现对运动控制卡IO的读写。6050运动控制卡有8个输出点，20个输入点，其中20个输入点中第9至第20点有双重目的，在缺省状态下，它们分别是4个轴的正限位输入点、零位输入点、负限位输入点。由于系统的X、Y、Z三轴需用到输入点的正限位输入点、零位输入点、负限位输入点来实现对轴位置的定位控制，所以系统对输入点的检测目前只检测1-8位。

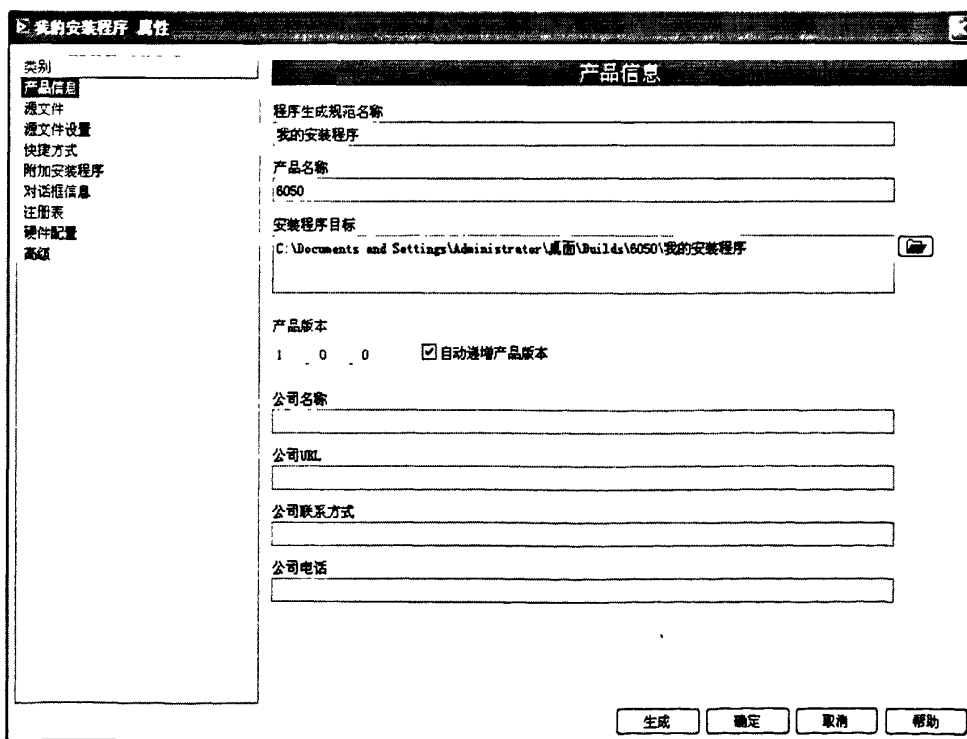
5.6 LabVIEW 程序打包

将LabVIEW程序打包的好处是能在没有安装LabVIEW软件的计算机上运行本程序,其前提是此计算机已安装LabVIEW Run-Time Engine,相对于庞大的LabVIEW软件,这样可以节省许多硬盘空间。其步骤如下:

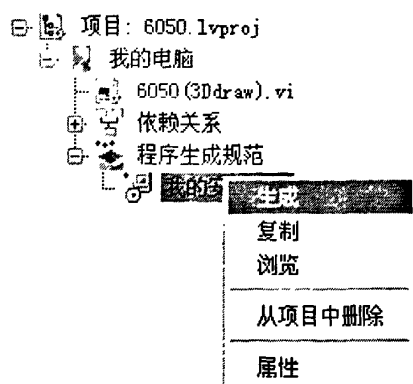
- 1、新建一个项目,把本程序添加进去并保存。



- 2、右键点击“程序生成规范”,选择新建生成安装程序,填写有关信息,把源文件添加进来并保存。



- 3、右键点击我的安装程序选择生成。



4、生成了安装程序还不可以，接下来就要生成应用程序，其过程基本上同生成安装程序一样，只是在右键单击“程序生成规范”时选择新建应用程序。

5、把安装程序安装在没有安装LabVIEW软件的计算机上，打开应用程序即可运行本程序。

5.7 本章小结

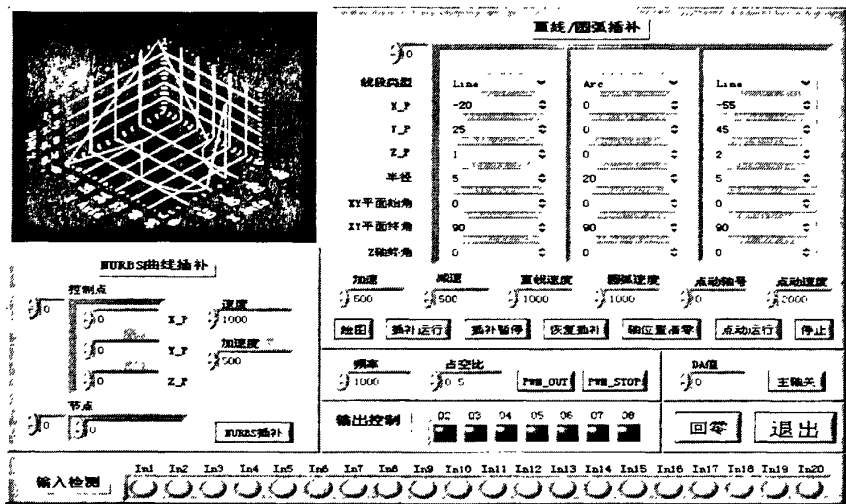
介绍了LabVIEW开发程序的步骤及人机控制界面的设计，研究了LabVIEW调用基于C语言编写动态链接的方法及建立了基于LabVIEW的控制卡驱动程序库，分析了运动轨迹仿真程序和控制程序的实现，。

第六章 系统验证

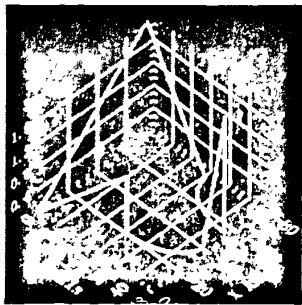
系统硬件和软件设计好后，其功能、精度是否达到预期的目标，需要通过实验来验证。本章将通过在有机玻璃上加工已仿真的图案来验证系统的功能和精度，具体如下所述。

6.1 仿真待加工的图案

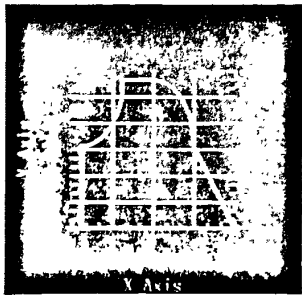
先选择“线段类型”，然后输入相应线段的参数，通过人机控制界面依次设置了6段曲线参数并单击“绘图”按钮仿真，如下图6-1所示，b)是其三维仿真图形，图中有空间直线、顺圆和逆圆等线段，c)是其在XY平面的投影图。



a)仿真控制界面



b)三维仿真图



c)XY 平面投影图

图 6-1 三维曲线仿真图

Figure6-1 Three-dimensional curve simulation diagram

每一段曲线的参数设置如下：

第1段，空间直线，起点为原点，终点为(-20, 25, 1)，在XY平面上投影的线段长度约为32.01562mm；

第2段，逆圆，半径 $R=20$ ，起始角度 $S_A=0$ ，终止角度 $T_A=90$ ；

第3段，空间直线，终点为(-55, 45, 2)，在XY平面上投影的线段长度为10mm；

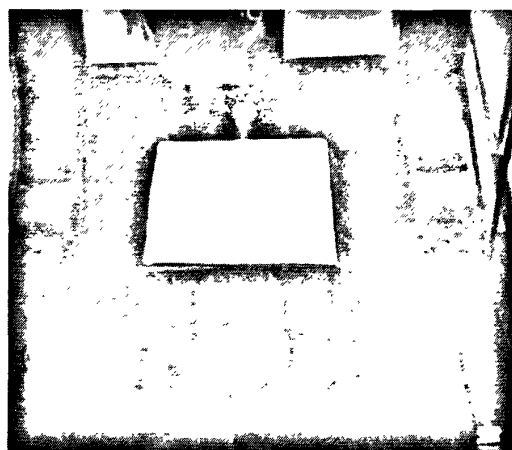
第4段，顺圆，半径 $R=20$ ，起始角度 $S_A=0$ ，终止角度 $T_A=-90$ ；

第5段，空间直线，终点为(-75, 0, 2)，在XY平面上投影的线段长度为25mm；

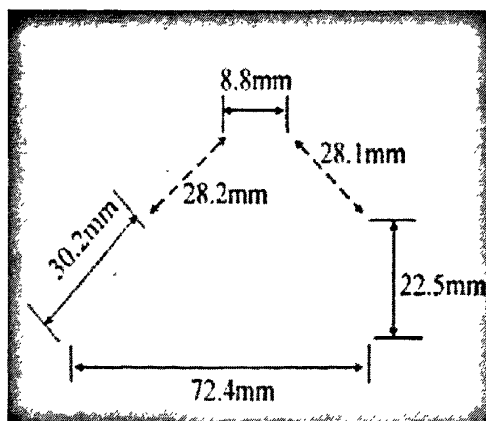
第6段，空间直线，终点为(0, 0, 0)，在XY平面上投影的线段长度为75mm；

6.2 加工已仿真的图案

将半径约为1mm的铣刀安装到电主轴，把有机玻璃固定在加工平台上，通过人机控制界面设定相应的速度、加减速度和主轴的转速，按下“回零”按钮，使铣刀回到系统的物理坐标原点，或使其移到要加工的位置起点，然后用“轴位置清零”方式，使其从当前位置开始加工，本次实验使用了后者，启动主轴，等主轴的转速达到了目标速度后，单击“插补运行”按钮，铣刀开始在有机玻璃上加工，加工好的图案如下图6-2所示。



a)加工有机玻璃



b)有机玻璃上加工的图案

图 6-2 实物加工图

Figure6-2 Material object processing diagram

6.3 实验分析总结

从实际加工出来的图案可看到共有 6 段曲线，每一段曲线的类型与仿真的一致，

即依次是直线、逆圆、直线、顺圆、直线、直线，并且每一段曲线的深浅不一样，达到了三维的效果，所以单从外形来看加工出来的图案达到了预期目标。

接下来需要对其进行精度分析，用尺子量度每一段曲线的长度如上图 6-2 中所标示，与仿真的每一段曲线长度对比如下表 6-1 所示：

表 6-1 精度分析

Table6-1 Precision analyses

线段	类型	仿真长度(mm)	实际长度(mm)	误差(mm)
1	直线	32.01562	30.2	1.81562
2	逆圆	28.28427(弦长)	28.2	0.08427
3	直线	10	8.8	1.2
4	顺圆	28.28427(弦长)	28.1	0.18427
5	直线	25	22.5	2.5
6	直线	75	72.4	2.6

从表中数据可看出，加工同类型的曲线产生的误差比较接近，且加工的曲线段越往后其产生的误差越大。通过仔细分析加工的过程，找出了产生误差的来源主要有以下 3 个：

1、由于铣刀在进给速度比较低，电主轴的旋转速度比较高时，铣刀与有机玻璃在摩擦过程中产生高温将有机玻璃的表面融化，融化的有机玻璃贴附在铣刀上，使铣刀的半径加大，这是产生误差的来源之一。

2、由于铣刀的半径约为 1mm，系统没有做刀具半径补偿，这是产生误差的来源之二。

3、系统方案使用的是开环控制系统，精度不够高，这是产生误差的来源之三。
为了提高加工精度，目前可以做的改进措施有：1、控制好铣刀的进给速度；2、增加系统的刀具半径补偿功能；3、使用闭环控制系统。

经过以上分析，系统的硬件及软件设计理论上是正确的，但还存在不足，以后需进一步改善。

6.4 本章小结

对比分析在有机玻璃上加工的图案与仿真的三维曲线，找出误差来源及提出了改进措施，验证了系统硬件及软件设计的正确性。

结论与展望

课题采用“PC+运动控制卡”构成多轴控制器的控制方案设计了基于LabVIEW的多轴运动控制系统，主要完成了以下工作：

1、研究和构建了三维控制系统，主要包括系统整体方案设计、控制卡结构及工作原理的研究、系统硬件连接测试等。

2、研究了常用的三种数值插补算法（逐点比较法、数字积分法和时间分割法）、加减速控制算法（梯形加减速、指数加减速和S型加减速）和NURBS曲线插补的原理特点，选择了适合于本系统的数值插补算法和速度控制算法。

3、基于LabVIEW设计了人机控制界面，开发了运动轨迹仿真程序和控制程序，并通过在有机玻璃上雕刻图案验证了系统整体方案设计的正确性。

由于本人的能力有限，再加上时间紧，工作量大，还有不完善的地方。在现有所做的工作基础上，今后还可以从以下几个方面作进一步研究，以进一步完善和提高基于LabVIEW的多轴运动控制系统的性能和精度：

1、将开环控制系统改为闭环控制系统。由于受到条件限制本系统开环的步进电机控制系统，存在一定的误差，为了提高系统的控制精度，可以采用闭环的伺服电机控制系统。

2、增加刀具半径补偿功能。由于时间紧，本系统没有做刀具半径补偿方面的研究，加工出来的图案与预期的图案存在一定的误差，为了提高系统的加工精度，要增加系统的刀具补偿功能。

3、扩大软件的适用范围。对市场上现有的板卡，特别是国产的中低端板卡进行研究，分别构建接口模块，组成通用的部分（一些上层模块）和针对不同厂家的特有部分（各板卡的接口模块），以进一步提高软件的适用性。

4、本系统目前以输入坐标的方式设置曲线，不支持G代码的读取，今后可从读取G代码方面进一步研究。

目前LabVIEW在数据采集和信号处理领域应用广泛，但在运动控制领域却应用不多，而且国内很少专家学者在这方面作研究，课题采用LabVIEW作为上位机控制软件开发了三维控制系统，具有一定的创新意义，为后续的应用开发奠定了基础。

参考文献

- [1] 刘冬敏,薛培军.浅谈运动控制技术的发展现状及应用前景[J].中州大学学报,2007,24(1):125-126.
- [2] 王家军,齐冬莲.运动控制系统的发展与展望[J].电气时代,2004,10:54-56.
- [3] Herrin,G.E.Open Modular Architecture Controllers(OMAC)[N]. Modern Machine Shop,1996,4.
- [4] Raymond E Chalmers.Open architerture CNC continue advancing[J].Manufacturing Engineering,2001,127(1):49-52.
- [5] 刘阳.虚拟仪器的现状及发展趋势[J].电子技术应用,1998(4):3-5.
- [6] 殷俊龄,刘新妹.测试系统虚拟仪器技术的研究与发展[J].山西电子技术,2002(2):17-19.
- [7] 吴琳,谭营,唐建.运动控制技术发展与展望[J].机床与液压,2007,35(7):232-233.
- [8] 周奉磊,马建伟.基于PCI总线的运动控制卡及其应用[J].世界仪表与自动化,2005,9(6):10.
- [9] 王淑芳.基于虚拟仪器技术的直流伺服电动机控制系统[J].电气时代,2006,11:58-59.
- [10] 刘日良,张承瑞.数控直线插补过程的一种新算法[J].组合机床与自动化加工技术,2002,3:1-3.
- [11] 陈黎融,白婕静,刘继伟.基于FPGA的数字积分法圆弧插补器的设计与实现[J].现代制造工程,2008,2:58-60.
- [12] 游达章,苏旭武,高贵兵,黄劲.两轴联动的数字积分插补算法的实现[J].装备制造技术,2008,1:41-43.
- [13] 张远.高速雕刻机数控插补技术的研究[D].南京:南京航空航天大学,2008.
- [14] Koren Y.Computer Control of Manufacturing Systems[M].New York:McGraw Hill,1983.
- [15] 荆伟.基于DSP的三次B样条曲线插补的实现[D].杭州:浙江大学,2007.
- [16] 梁静.数控机床插补技术的研究[D].沈阳:沈阳工业大学,2007.
- [17] 许良元,桂贵生,彭丹月.高速加工中的加减速控制[J].CAD/CAM与制造业信息化,

- 2005,26(9):66-67.
- [18] 毕承恩,丁乃建.现代数控机床[M].北京:机械工业出版社,1993.
- [19] 杜君文,邓广敏.数控技术[M].天津:天津大学出版社,2002.
- [20] 卢胜利,王睿鹏,祝玲.现代数控系统——原理、构成与实例[M].北京:机械工业出版社,2006.
- [21] Yeh S.S,Hsu P.L.Speed-controlled interpolator for machining parametric curves[J]. CAD Computer Aided Design,1999,31(5):35-36.
- [22] De Souza A.F,Coelho R.T.Experimental investigation of feed rate limitations on high speed milling aimed at industrial applications[J].International Journal of Advanced Manufacturing technology,2007,32:12-13.
- [23] 张玉姣.基于DSP的运动控制器的设计与研究[D].武汉:华中师范大学,2009.
- [24] 冯寿廷.构件化的运动控制系统设计、实时性分析以及优化设计方法研究[D].广州:华南理工大学,2006.
- [25] 吴凡,杨小军,曹剑中,刘波.基于DSP+CPLD的高速高精度PCI运动控制卡[J].弹箭与制导学报,2006,26(2):1030-1032.
- [26] PLX Technology Inc.PCI 9052 Data Book[M].USA: PLX Technology Inc,2001.
- [27] 赵政春,邱飏.ATP系统伺服平台的研究[J].机床与液压,2009,37(6):121-123.
- [28] 武友德,胡明华,李登万.基于负逻辑控制的数控机床限位保护装置及方法研究[J].制造技术与机床,2007,2:64-66.
- [29] 高建设,陶征,程丽,苏宇峰.新型五自由度并联机床回零研究[J].机床与液压,2009,37(6):7-9.
- [30] 任晓虹,吴芳,孙建业.数控机床的回零及应用[J].机械设计与制造,2009,5:193-194.
- [31] 刘喜梅,周春香.PLC控制系统在工业应用中的抗干扰分析[J].测控技术,2009,28(2):96-98.
- [32] 熊轶娜,吴跃明,陈洁.数控机床控制系统的抗干扰分析[J].控制与检测,2009,8:65-67.
- [33] 赵国伟,徐跃民.基于VB和LabVIEW的自动化控制和编程设计[J].微计算机信息,2007,23(11):94-95.
- [34] 王淑芳.基于虚拟仪器技术的多轴步进电机控制系统[J].微计算机信息,2006,22(9):97-99.

- [35] 杨乐平,李海,涛杨磊.LabVIEW程序设计与应用[M].北京:电子工业出版社,2005.
- [36] Truchard Jim.Build your own virtual instrument[J].Canadian electronics,1996,11:12-13.
- [37] Nations Instruments.LabVIEW User Manual[Z]. USA:NI,2000.
- [38] National Instrument Corporation.G Programming Refeerence Manual[Z]. USA:NI, 1999.
- [39] 齐晓辉,林茂六,王重阳.LabVIEW平台功能扩展方法——CIN技术研究[J].测控技术,2005,24(11).
- [40] National Instruments.Using External Code in LabVIEW7[Z]. USA:NI,2003.
- [41] National Instruments Corporation.LabVIEW Foundation and VI Reference Manual[Z]. USA:NI,1998.
- [42] 刘玉.利用动态连接库扩展LabVIEW的功能[J].国外电子测量技术, 2004:51-53.
- [43] Fountain,Tim.LabVIEW moves into the real-time domain[J].Electronic Engineering Design,1999,1:42-43.

攻读硕士学位期间发表的论文

1. 梁永忠, 高军礼, 邓则名. 基于LabVIEW的直线电机监测系统设计. 机床与液压. (已录用)
2. 梁永忠, 高军礼, 邓则名. 基于LabVIEW的三维运动轨迹仿真程序设计. (拟投稿)

独创性声明

秉承学校严谨的学风与优良的科学道德，本人声明所呈交的论文是我个人在导师的指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，不包含本人或其他用途使用过的成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明，并表示了谢意。

本学位论文成果是本人在广东工业大学读书期间在导师的指导下取得的，论文成果归广东工业大学所有。

申请学位论文与资料若有不实之处，本人承担一切相关责任，特此声明。

论文作者签字：梁永志

指导教师签字：邓明、高军

2010年6月9日

致 谢

值此论文完成之际，谨向在这三年来所有曾给予过我帮助的老师、同学及亲友们致以我最诚挚的谢意！

衷心感谢邓则名教授、高军礼副教授，本论文是在他们的悉心指导下完成的。两位导师在学习、工作、生活等多方面给予我诸多的帮忙和和亲切关怀。同时他们对学生认真负责、严谨求是的态度、勤勉的工作作风和勇于创新的精神值得我终身学习。在毕业之际，谨向邓老师、高老师表达我最诚挚的敬意和最衷心的感谢。

衷心感谢华南理工大学的宋海涛博士在专业学习、论文研究和论文撰写等方面给予我无私的指导和帮助。

衷心感谢同学胡妙璇、谢慧波、彭逢彪、陈世军，师弟权宁辉、刘汉东、张英范、杨洪天、冯东英以及实验室的其他同学和师弟、师妹给我的支持和帮助。

最后感谢我的家人对我一如既往的支持、关怀，使我勇敢地面对挫折，勇往直前！谨以此文表达我最诚挚和深切的谢意。

基于LabVIEW的多轴运动控制器应用研究



作者：[梁永忠](#)
学位授予单位：[广东工业大学](#)

本文链接：http://d.g.wanfangdata.com.cn/Thesis_Y1745696.aspx

授权使用：广东工业大学图书馆(gdgydxtsg)，授权号：8f70ea5a-918c-4793-8c17-9ebf00bbcada

下载时间：2011年4月8日